Mixing, transport and turbulence modulation in solid suspensions : study and modelling

François Laenen

To cite this version:

François Laenen. Mixing, transport and turbulence modulation in solid suspensions : study and modelling. Other [cond-mat.other]. COMUE Université Côte d'Azur (2015 - 2019), 2017. English. NNT : 2017AZUR4010 . tel-01534441

HAL Id: tel-01534441
https://theses.hal.science/tel-01534441v1

Submitted on 7 Jun 2017

Université Côte d'Azur
École doctorale de Sciences Fondamentales et Appliquées

Unité de recherche :
Laboratoire Joseph-Louis Lagrange – UMR 7293

Thèse de doctorat

Présentée en vue de l'obtention du
grade de Docteur en Sciences de
Université Côte d'Azur

Discipline : Physique

Présentée et soutenue par
François Laenen

Mixing, transport and turbulence modulation in solid suspensions

Study and modelling

Dirigée par *Jérémie Bec, Directeur de Recherche CNRS, Observatoire de la Côte d'Azur et codirigée par Giorgio Krstulovic, Chargé de Recherche CNRS, Observatoire de la Côte d'Azur*

Soutenue le 24 février 2017

Devant le jury composé de :

Sergio Chibbaro	Maitre de Conférence HDR, Université Pierre et Marie Curie	Rapporteur
Romain Volk	Maitre de Conférence HDR, ENS Lyon	Rapporteur
Guido Boffetta	Professeur, Université de Turin, Italie	Examinatrice
Aurore Naso	CR CNRS, Ecole centrale Lyon	Examinatrice
Mikhael Gorokhovski	Professeur, Ecole centrale Lyon	Examinateur
Emmanuel Villermaux	Professeur, Aix-Marseille Université	Examinateur
Jérémie Bec	DR CNRS, Université Côte d'Azur	Directeur
Giorgio Krstulovic	CR CNRS, Université Côte d'Azur	Co-directeur

Summary

The transport of particles by turbulent flows is ubiquitous in nature and industry. It occurs in planet formation, plankton dynamics and combustion in engines. For the dispersion of atmospheric pollutants, traditional predictive models based on eddy diffusivity cannot accurately reproduce high concentration fluctuations, which are of primal importance for ecological and health issues.

The first part of this thesis relates to the dispersion by turbulence of tracers continuously emitted from a point source. Mass fluctuations are characterized as a function of the distance from the source and of the observation scale. The combination of various physical mixing processes limits the use of fractal geometric tools. An alternative approach is proposed, allowing to interpret mass fluctuations in terms of the various regimes of pair separation in turbulent flows.

The second part concerns particles with a finite and possibly large inertia, whose dispersion in velocity requires developing efficient modelling techniques. A novel numerical method is proposed to express inertial particles distribution in the position-velocity phase space. Its convergence is validated by comparison to Lagrangian measurements. This method is then used to describe the modulation of two-dimensional turbulence by large-Stokes-number heavy particles. At high inertia, the effect is found to be analogous to an effective large-scale friction. At small Stokes numbers, kinetic energy spectrum and nonlinear transfers are shown to be modified in a non trivial way which relates to the development of instabilities at vortices boundaries.

Résumé

Le transport de particules par des écoulements turbulents est un phénomène présent dans de nombreux écoulements naturels et industriels, tels que la dispersion de polluants dans l'atmosphère ou du phytoplancton et plastiques dans et à la surface des océans. Les modèles prédictifs classiques ne peuvent prévoir avec précision la formation de larges fluctuations de concentrations.

La première partie de cette thèse concerne une étude de la dispersion turbulente de traceurs émis à partir d'une source ponctuelle et continue. Les fluctuations spatiales de masse sont déterminées en fonction de la distance à la source et à l'échelle d'observation.

La combinaison de plusieurs phénomènes physiques à l'origine du mélange limite la validité d'une caractérisation de géométrie fractale. Une approche alternative est proposée, permettant d'interpréter les fluctutations massiques en terme des différents régimes de séparation de pair dans des écoulements turbulents.

La seconde partie concerne des particules ayant une inertie finie, dont la dispersion dans l'espace des vitesses requiert de développer des techniques de modélisation adaptées. Une méthode numérique originale est proposée pour exprimer la distribution des particles dans l'espace position-vitesse. Cette méthode est ensuite utilisée pour décrire la modulation de la turbulence bi-dimensionnelle par des particules inertielles. A grand nombres de Stokes, l'effet montré est analogue à celui d'une friction effective à grande échelle. Aux petits Stokes, le spectre de l'énergie cinétique du fluide et les transferts non-linéaires sont modifées d'une manière non triviale.

Contents

Nomenclature

$\boldsymbol{V}$	Tracer velocity.
$\boldsymbol{V}_p$	Inertial particle velocity.
$\boldsymbol{X}$	Tracer position.
$\boldsymbol{X}_p$	Inertial particle position.
$\boldsymbol{x}_S$	Position of the injecting source.
$\delta_r u$	Longitudinal velocity increment between two particles separated by a distance r.
ϵ	Total energy dissipation rate.
ϵ_α	Energy dissipation rate due to large-scale friction.
ϵ_ν	Energy dissipation rate due to viscous friction.
κ	Molecular diffusion coefficient.
κ_T	Turbulent diffusion coefficient.
κ_V	Diffusivity in the velocity space.
ϕ_m	Massic ratio between the solid and the fluid phase.
ϕ_S	Source injection rate. Units are T^{-1}.
ϕ_v	Volumic fraction of the particles.
ψ	Fluid stream function.

$\rho(\boldsymbol{x})$	Fluid density field.
$\rho_p(\boldsymbol{x})$	Particle density field. $\rho_p(\boldsymbol{x}) = n_p(\boldsymbol{x})M_p$ where M_p is the mass of one particle. Units of ρ_p are ML^{-d}, where d is the space dimension.
u_{rms}	Root-mean square value of the velocity field, averaged over all components.
a	Particle radius.
d	Dimension of the position space.
m_p	Total mass of the particle population.
m_{QL}	Quasi-Lagrangian mass.
$n_p(\boldsymbol{x})$	Particle number density field. Units are L^{-d}, where d is the space dimension.
pdf	Probability density function.
$R_{\max}$	Radius of the circle around the injecting source beyond which particles are removed from the domain.
St	Stokes number.
T_{C_L}	Lagrangian velocity autocorrelation time.
Z	Fluid enstrophy.

CHAPTER 1

Introduction and context

This thesis is to be put in the context of the ERC research project *Atmoflex*, running from 2010 to 2014, from which it was partly funded. This project aimed at providing a better understanding of fluctuations in particulate transport and mixing processes in turbulent flows, such as pollutants in the atmosphere or scalar fields like salinity in the oceans, or merging density-matched gases.

Turbulence: a multi-scale phenomenon

Turbulence is a phenomenon characterised by a chaotic and out of equilibrium state of a physical non-linear system. It may be found in numerous situations and applications, such as non-linear optics, passive advection in fluids, waves interactions at water surface, magnetic dynamos, etc. All these processes involve a system in which energy is exchanged between many degrees of freedoms. Some theories emerged during the last century trying to find rigorous mathematical formulation of the phenomenological predictions with the use of non-equilibrium statistical mechanics, but an unifying and universal theory is still lacking.

In this thesis is considered the Navier–Stokes turbulence, recognisable by the broad range of vortex sizes it generates in fluids. It has the particularity of re-distributing energy injected at a scale L down to smaller and smaller scales until molecular dissipation stops this *cascading* energy transfer. This transfer can be thought of as a pipe carrying energy in the scale space. It takes place in a range of scales, called *inertial range*, in which statistics of the flow are believed to be independent of the way it is forced and dissipated. This transfer arises because of the non-linear term in the Navier–Stokes equations which makes this equation unsolvable exactly.

A striking characteristic of turbulent flows is the apparent chaotic trajectories of the species they transport. This manifests into the unpredictability of the position of solid grains in suspensions, or concentration values downstream of an emitting source of pollutant. The same observation holds for the simultaneous transport of multiple objects. Consider for example placing two buoys on the surface of a river in a turbulent state. initially very close to each other, then compare their trajectories. Whatever small are their initial separation, they are likely to diverge in a finite time, demonstrating the *chaotic* nature of turbulence through the sensibility to initial conditions.

Even though chaos theory underwent tremendous interest since the beginning of the 20^{th} century, the statistics of the velocity differences between two points in space in the inertial range lead to pair separation rates different from what can be predicted by chaotic motions. This mixing of separation regimes between particles lead to even more complicated prediction about their average concentration with possibly high fluctuations.

In addition, there still subsists a lack of understanding regarding the *universality* of the fluid velocity statistics, i.e. the independence with respect to the forcing and dissipation mechanisms. One is then forced to treat virtually each situation as a case-by-case study.

Lagrangian and Eulerian description

Considering the release of a cloud of a given substance, such as dye or passive pollutant, in the atmosphere or in the ocean, one may ask multiple questions: how will its shape be deformed with time? What would be the maximum expected concentrations? What about the strength of its variations in space and time ? What are the probabilities that a concentration grows above a given threshold and how often?

These questions are naturally raised by health and environmental issues (Shi *et al.*, 2001). Indeed, it is often required to predict concentration levels of various constituents, like ash, particulate matter, radioactive elements, etc. Respective examples are volcanic eruptions and their impact on air traffic (Schäfer *et al.*, 2011), road traffic regulation in periods of intense air pollution (Han & Naeher, 2006), intra and intercontinental radioactive transport (Wotawa *et al.*, 2006), etc. Living organisms are also concerned by large fluctuations: for the ones following concentration gradients (*chemotaxis*), such as moths attracted by male pheromones, (Mafra-Neto *et al.*, 1994), large scale-induced density fluctuations make this approach much more difficult.

Furthermore, the multi-scale property of turbulent flows may be appreciated into its *self-similar* character: irregular patterns made by smoke coming out of a cigarette resemble the ones escaping from a small house chimney, or a large industrial power plant, or even, to some extents, from a volcano. Another noticable effect of turbulent transport is the fact that particles may get trapped inside vortices, leading to higher concentration values. Actually, it is known that the probability density function of the passive scalar concentration have tails decreasing slower than a Gaussian distribution (Warhaft, 2000). Such trapping events may then be related to regions of the flow with given topological properties (see, for instance, Bhatnagar *et al.* (2016)). Figure 1.1 shows some examples of systems concerned with transport of solid particles or continuous fields .

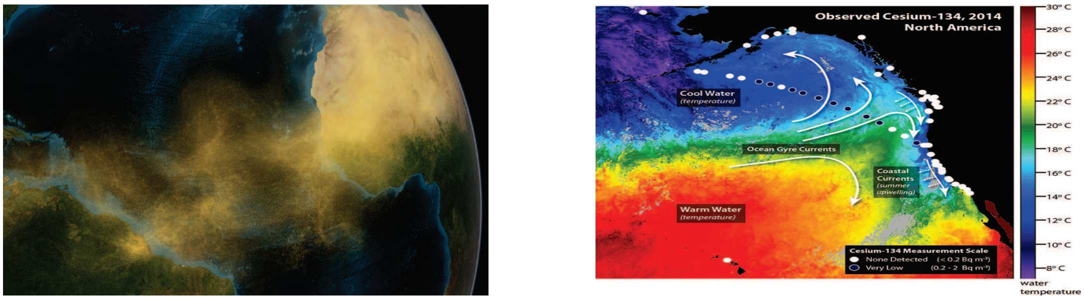
When measuring the spatial and temporal properties of the transported species, one can choose to adopt two main points of view. The first one is *Lagrangian* (Lin *et al.*, 2011). It forces the description of phenomena in a framework attached to the transported particle. Lagrangian models are very useful for various reasons. They offer a more natural way to model turbulent transport of solid particles. In addition, they suffer from only infinitesimal numerical diffusion, allowing to recover strong concentration gradients. They are also more numerically stable and allow for bigger time steps. Furthermore, the inverse Lagrangian transport may be used to track sources of contaminants or green house gases (Trusilova *et al.*, 2010). Finally, computational resources available nowadays allow to routinely simulate systems with millions of particles, which is sufficient in some application, and was undoable a decade ago.

Another way to measure turbulent transport is by considering a continuous concentration field of a given released substance. We then talk about *Eulerian* description. This point of view considers a fixed spatial grid on which are defined tensorial quantities, mostly scalars, like concentration. Although Eulerian framework suffers from numerical diffusion and instabilities, it offers a more natural description of a field transported by the underlying carrier flow, and allows for more convenient way to model back reaction from the substance on the fluid. Eulerian formulations also offer more handy ways to parallelise numerical codes, especially when considering domain decomposition among a large number of computational nodes.

The challenge of numerical simulations

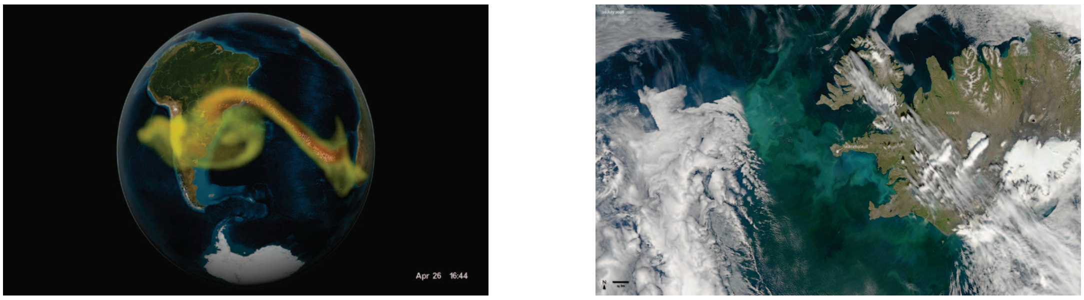
In order to perform fundamental studies regarding all questions risen above, computers have played a growing role into fluid mechanics since their invention in the middle of the 20^{th} century, although methods of discrete calculation to resolve the fluid equations already appear in the pioneering works of Richardson (see Richardson (2007) (reprint) and reference in Hunt (1998)). First computations based partly on these works appeared during the 40s using ENIAC and the first three-dimensional simulations were later performed Los Alamos laboratory (Harlow, 2004). Since then, more and more power was dedicated to simulate flows as realistic as possible.

One could ask why tremendous supercomputer that we see today in 2016 are still not sufficient ? This is again due to the multi-scale nature of turbulence but also to its temporal fluctuation. Indeed, turbulence is characterised by different eddy sizes *and* time scales. The width of the spatial scale separation can be measured by the adimensional Reynolds number Re, which is the ratio between advection strength UL and viscosity κ. The larger Re, the wider the scale separation. Furthermore, the complexity of numerical calculations, or number of degrees of freedom, grows as $Re^{9/4}$, which translates into a rapid increase of the need for computational resources.



(a) Amazonia forest fed with transported sand from Sahara desert. See Yu *et al.* (2015).

(b) Surface sea temperature along with concentration spots of radioactive Cesium 134 following Fukushima eruption. Credit: WHOI



(c) 3D simulation of ash spreading following the Calbuco eruption in April 2015 combined with actual data from Suomi NPP satellite. Credit: NASA.

(d) Phytoplankton bloom off the Iceland coast. Credit: NASA.

Figure 1.1: Examples of relevant issues involving turbulent transport.

Turbulent motions often arise above a critical Reynolds number, of the order of a few thousands, depending on the system. Very high Reynolds numbers are especially found in the case of planetary-scale motions, where the separation in scales spans from millimetres to kilometres, Re can reach 10^8. For a moving car at 90 km/h, it reaches 10^6. Furthermore, because of its inherent *intermittent* character (see section 2.1.1), i.e. with velocity statistics displaying tails much broader than Gaussian, turbulence is also characterised by extreme events. These can represent up to 10^5 times the mean value. They are not so probable (hence the name extreme) but sufficiently to play a key role in the dynamics, a property shared with other strongly non-linear phenomena. Because of these extreme events, numerical simulations have to be run for a very long time and / or at very large resolutions.

State of the art three-dimensional turbulence numerical simulations achieve Reynolds numbers up to 45000, with a Taylor microscale Reynolds of 1300 for a resolution of 8192^3 grid points (Yeung *et al.*, 2015).

How to overcome this numerical challenge and to reduce its complexity ? The answer is not to resolve explicitly all the scales. Tons of scientist and engineers have worked in this direction, leading to very clever and tricky methods and mathematical models to represent the turbulent motions at scales smaller than the one of interest. Such activities are referred to as *sub-grid scales* modelling (Majda & Kramer, 1999; Pope, 2000). Such small scale motions however play a very important role, impacting larger ones in situations like clouds (Bodenschatz *et al.*, 2010), or planets (Armitage, 2015) formation. Careful design of the models is thus required, and these must be continuously improved based on a better knowledge of microscopic phenomena.

Particle transport is a typical situation where such problems occur. Traditional estimations used in the mechanical and environmental engineering communities are based on *mean-field* approach (Opper & Saad, 2001): substances are advected by larger scales (big eddies) and the effect of smaller scales are just perturbations, treated for example as additional sources of diffusion, called *eddy-diffusivity*. Even if we know that the effect of small scales motions on the larger ones cannot be accurately represented by a simple diffusion operator (Corrsin, 1975). These models yield correct predictions for concentrations that are far from an emitting source or during long time averages.

This topic is itself enclosed into a much wider research area, *multiphase flow modelling*, dealing with the numerical simulation of multiple species simultaneously present in a spatial domain, interacting or not. An example of application is the prediction of the fluid regimes transition (Labourasse *et al.*, 2007; Monahan & Fox, 2007; Van der Hoef *et al.*, 2008), when one wishes to test how a device such as a fluidized-bed reactor will scale when going from the laboratory to the power plant (Ge *et al.*, 2007).

When back-reactions come at play

Under some circumstances, the transported phase may have a significant impact on the carrier flow, which may be desirable or not. Such effects are called *two-way coupling*, and induce non-linear effects that can hardly be predicted by phenomenological arguments. In the case of dispersed solutions, several application are worth mentioning.

Consider the example of fluid transportation in a pipe. Pressure losses in such conducts lead to very high power consumption. It is thus of economical interest to try to reduce energy dissipations in such flows. This dissipation attenuation was actually observed in suspensions of certain additive types. Fibrous additives, like polymers (e.g. nylon, cotton) have shown to reduce drag efficiently (White & Mungal, 2008; Yang, 2009) though the situation is not yet clear for non fibrous materials, like rigid bodies of various shapes (spherical or platelet, needle-shaped...) and size. The interplay between suspended solids and the liquid phase is also a process of prime importance for planet formation (Barranco & Marcus, 2005). Indeed, gas giant planets have been shown to migrate, i.e., to form far from their star and come closer at later stages, with the corresponding mechanisms depend on their size.

The modelling of these back reactions leads to additional challenges. Indeed, when using Lagrangian description, one needs to reconstruct a force field to act on the carrier phase, and losses of convergence and precision occur a this step. When available, an Eulerian point of view for the suspended phase is thus generally preferable.

The case of two dimensional turbulence

Turbulent flows that are considered in this thesis are two dimensional. Actually, there is no such a thing as a real two-dimensional flow in nature, so why bother studying them ?

In some situations, a flow which is *a priori* three dimensional may exhibit $2D$ turbulence dynamics. A particular example is the case when swirling motions are *physically constrained* to evolve in a thin layer. These constrains may take various forms. They can be created through the formation of a thin layer by geometric confinement with solid boundaries, or by the presence of strong rotation (Pouquet *et al.*, 2013) and / or stratification, in which case the formation of two dimensional layers called *pancakes* is observed (Godoy-Diana *et al.*, 2004). Some systems may combine all these effects. Examples are large-scale geophysical flows in the Earth atmosphere and oceans (Nastrom & Gage, 1985; Monin & Ozmidov, 1985). One way to reproduce such constrains in laboratory experiments is to isolate the fluid within a thin tank. One can then use electrically conducting fluid put into motion by an array of magnets (Paret *et al.*, 1999; Boffetta *et al.*, 2005). Soap films are also a really good ”planetary toy” because this setup allows to reproduce turbulence under gravity in a thin spherical layer combined with rotation (Kellay *et al.*, 1998; Rutgers, 1998; Seychelles *et al.*, 2010).

Another notable example of flows dimensionally constrained are the protoplanetary disks (Barranco & Marcus, 2005) where tall columnar vortices form. In gaseous nebulae, both processes of planetary formation and migration depend on the vortices structure. Accretion probabilities and transport will indeed vary depending if the flow is dominated by turbulent eddies and long-lived coherent vortices.

One should recover three dimensional turbulence phenomenology when considering the constrained flow at sufficiently small scales, or when removing the constrains. For example, dimensionality may be measured as a function of the rotation or stratification intensity (Smith *et al.*, 1996; Deusebio *et al.*, 2014; Sozza *et al.*, 2015). This mechanisms at transition between $2D$ and $3D$ is called *bidimensionalisation* and is also a very interesting research topic by itself.

All the notional concepts described above form the core part of this thesis work. How does the temporal and spatial correlations of fluid velocity in two dimensional turbulence impact mass distribution when it is initially released from a limited region in space ? How is the turbulence affected by small inertial particles ? How to represent such particles in term of a field in a way that is computationally affordable and physically correct ?

This manuscript is organised in two parts. The first one is aimed at describing how mass continuously injected from a point source in two-dimensional turbulence fills the space, targeting situations such as oil spreads at ocean surface. Lagrangian particles are emitted via numerical simulations, and mass fluctuations are quantified using fractal dimension and a novel description based on relative pair separation.

The second part is split in two chapters. The first one describes a numerical approach for the simulation of small heavy particle suspensions in two-dimensional turbulence. It treats the kinetic equation associated to the dynamic on a regular lattice in position space and finite volume method in velocity space. The second chapter shows an application to the study of turbulence modulation by small and heavy particles. The modifications of large and small scales statistical quantities of the fluid are assessed.

In order to realise these studies, scientific libraries have been developed for massively parallel computations using GPGPUs. These libraries were used to simulate the two dimensional turbulence from Navier–Stokes equations as well as particle emission and dynamics. Some details about their implementation are explicited in the appendix A. Another library to handle systems of large number of particles was jointly developed. It was aimed to ease the process of creating particles with various properties like mass, electric charge, etc. while keeping performance when simulating their dynamic on various parallel architectures (see B).

The two parts may be read independently, with some concepts being introduced in chapter 2.

First part: Tracers dispersion from a point source

One can have the intuitive picture that on average, an emitted puff of a suspended substance will regularly grow under the effect of diffusion and spread uniformly in space. However, the temporal correlation of the turbulent eddies at all scales bringing together regions of very different concentration values results in creating strong inhomogeneities and gradients. In chapter 3, a system of tracer particles continuously emitted from a point source is studied. The additional challenge compared to traditional turbulent mixing lies in the joint effect of spatial as well as temporal correlations in the particles trajectories. The interplay of these correlations is one of the major issues in turbulence. Only in very few models for the carrier fluid these correlations can be analytically treated, like in the Kraichnan ensemble where temporal correlations are fully disregarded (Celani *et al.*, 2007). High resolution direct numerical simulations of inverse turbulent energy cascade are carried, and the issue of measuring the spatial fluctuations of the particle distribution is addressed. To this end, we propose a phenomenological description which allows us to relate the concentration fluctuations along particle trajectories (*quasi-Lagrangian mass scaling*) with the tracers

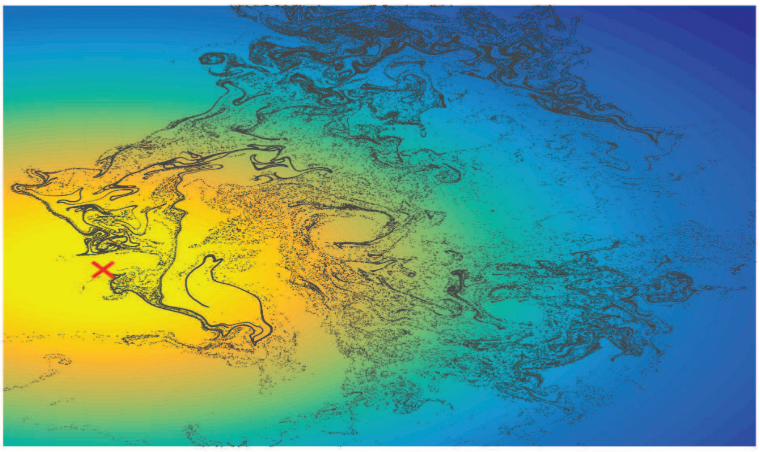


Figure 1.2: Gaussian concentration distribution (colour map) predicted by mean-field approaches do not allow to reproduce fine structures and high concentration levels of transported particles (black dots). The red cross represents the emitting source.

relative dispersion regimes. The idea is to follow an emitted line of particles to quantify its foldings and see how it contributes to the quasi-Lagrangian mass scaling as a function of the distance from the source.

Second part: Modelling particle-laden flows and two-way coupling

Challenges of inertial particles statistical modelling

This second part starts with chapter 4 which addresses the issue of modelling heavy-particle-laden turbulent flows. The dynamics of such particles is first introduced, and the challenge is stressed to provide a correct modelling of this kind of suspensions, which basically lies in the capacity of the particles to form caustics. The adopted mathematical model must then be able to resolve the velocity dispersion of the particle population. A short review of multiphase models dealing with solid suspensions is also presented.

A novel numerical method is then introduced. Its originality lies in the absence of any form of closing of the kinetic equation associated with the dynamic. This Liouville equation is integrated explicitly in the phase space. Numerical and physical convergence are assessed, and it is shown that the method reproduces with good accuracy the particle distributions obtained via Lagrangian direct simulations.

Turbulent modulation by small heavy particles

Chapter 5 presents a study about the impact of back-reaction of small heavy particles on a two-dimensional turbulent flow. A short overview of these effects is first presented and the chosen models representing the considered Stokes number asymptotics are introduced. Direct numerical simulations of direct enstrophy cascade are performed, varying the particles mass load. The effect of particles on various statistical properties of the flow is assessed in the asymptotics of low and large Stokes numbers. The modification of global quantities is measured, such as mean energy and enstrophy, as well as the modifications of the scaling in the velocity field through non-linear transfers and dissipations. Impacts on small-scale statistics is also addressed, in particular the modification of intermittency modification. Finally, we also measured how the particles preferential concentration property is affected.

CHAPTER 2

Definitions and concepts

Contents

Concepts and mathematical tools relevant to each chapter will be introduced in their respective opening. As all chapters of this thesis share in common the framework of two-dimensional, incompressible turbulence, the characteristics of such flows are highlighted and compared to their three-dimensional equivalent in this chapter. Some concepts and notations regularly appearing throughout this manuscript are also introduced.

2.1 Navier–Stokes equations

An incompressible velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ at position $\boldsymbol{x}$ is described by the following equations:

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}, \tag{2.1}$$

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

ν is the kinematic viscosity, the term $\nu\nabla^2$ being responsible for the dissipation of large gradients at small scales. p is the pressure which ensures the incompressibility condition (2.2). $\boldsymbol{F}$ denotes the external forcing that maintain the velocity field in a statistically steady, developed turbulent state.

In the absence of dissipation and forcing, i.e. when $\nu = 0$ and $\boldsymbol{F} = \boldsymbol{0}$, several quantities, or *invariants* characterise the flows. In three dimensions, global invariants are energy $E = \left\langle \|\boldsymbol{u}\|^2 \right\rangle$ and helicity $H = \langle \boldsymbol{\omega} \cdot \boldsymbol{u} \rangle$. The averaging operation is taken over space and time in the statistically stationary regime.

Properties of the flow may be assessed through the *rate-of-strain tensor*, a second order tensor encompassing the gradients of each velocity component $u_{1,2,3}$, namely:

$$\mathcal{A} = \boldsymbol{\nabla u} = \begin{pmatrix} \partial_1 u_1 & \partial_2 u_1 & \partial_3 v_1 \\ \partial_1 u_2 & \partial_2 u_2 & \partial_3 v_2 \\ \partial_1 u_3 & \partial_2 u_3 & \partial_3 v_3 \end{pmatrix}.$$

This matrix, like any, can be decomposed into a symmetric and an antisymmetric part, which are often named $\boldsymbol{\Omega}$ and $\boldsymbol{S}$, respectively:

$$\boldsymbol{\Omega} = \frac{1}{2}\left[\boldsymbol{\nabla u} - (\boldsymbol{\nabla u}))^T\right], \qquad \boldsymbol{S} = \frac{1}{2}\left[\boldsymbol{\nabla u} + (\boldsymbol{\nabla u}))^T\right]. \tag{2.3}$$

$\Omega_{ij} = \frac{1}{2}(\partial_i u_j - \partial_j u_i)$ are the vorticity components, and $S = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ the shear components.

From these tensors, a criterion can be built to determine whether locally in space the flow is dominated by shear or vorticity. In two dimensions, this criterion is given by the Okubo-Weiss criterion $\mathcal{W} = \Omega^2 - S^2$ (Okubo, 1970; Weiss, 1991). Due to the Poisson equation for pressure $\nabla^2 p = \Omega^2/2 - S^2$, the following relation holds for incompressible flows: $\left\langle \Omega^2 \right\rangle = 2\left\langle S^2 \right\rangle$.

In three dimensions, one can use the $Q-R$ criterion. The matrix $\mathcal{A}$ has three invariants under canonical transformations: $P = Tr(A)$, $Q = -Tr(A^2/2)$, $R = -Tr(A^3/3)$. The determinant of the characteristic equation for $\mathcal{A}$ is then given by $\Delta = (27/4)R^2 + Q^3$ and the $Q - R$ plan defines 4 regions corresponding to different combinations of eigenvalues of $\mathcal{A}$. Vorticity dominates for large positive Δ with vortices that are either compressed ($R < 0$) or stretched ($R > 0$). On the contrary, strain will dominate for $\Delta < 0$. See Cantwell (1993) for more details.

2.1.1 Structure functions and intermittency

The statistics of velocity differences between two points separated by a distance r constitute an important quantity in turbulent flows. Their probability density function gives an important idea about collision probabilities or particles separation rate (see section 2.3). The scaling of their moments also gives information about the scale-invariance of turbulent flows. Velocity differences, or *increments*, are defined by:

$$\delta_r \boldsymbol{u} = \langle \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r}) \rangle \,. \tag{2.4}$$

They constitute a random quantity. When considering homogeneous, isotropic flows, as is the case in this manuscript, $\delta_r \boldsymbol{u}$ only depends on the modulus of $\boldsymbol{r}$ and is independent of the position $\boldsymbol{x}$. The structure functions are defined by its moments of various order:

$$S_q(r) = \left\langle (\delta_r^{||} u)^q \right\rangle \tag{2.5}$$

where $\delta_r^{||} u = \delta_r \boldsymbol{u}(\boldsymbol{r}) \cdot \frac{\boldsymbol{r}}{r}$ is the *longitudinal* component of $\delta_r \boldsymbol{u}$. Alternatively, one may also consider $S^{\perp}$ using the projection of $\delta \boldsymbol{u}(\boldsymbol{r})$ on directions orthogonal to $\boldsymbol{r}$.

The average in (2.5) is taken over the whole space and over time series in the statistically steady regime. In a developed turbulent regime, $S_q(r)$ follows a power-law function of the scale r:

$$S_q(r) \propto r^{\zeta(q)} \tag{2.6}$$

The scaling behaviour is the following. Suppose a large scale forcing at l_I of the flow which is in a statistically stationary state. At scales smaller than l_I, statistical quantities can be assumed to be homogeneous. This is the *inertial range*. This regimes goes down to a scale l_d, the *dissipative scale*, at which energy is dissipated. In 1941, Andrei Kolmogorov made a series of hypothesis that leads to quantitative predictions about velocities increments $\delta_r u$ and l_d. One is that the energy dissipation rate $\epsilon > 0$ (we choose here to consider this quantity as positive) has a non-zero limit at vanishing viscosity ($\nu \to 0$). This is called *dissipative anomaly*. Another of his hypothesis, shared with Onsager and Heisenberg, sometimes called *universality assumption*, is that all the small-scale statistical properties are uniquely and universally determined by the scale r and the energy dissipation rate ϵ. Given the dimensions of the relevant quantities at play in the inertial range, $[\delta_r u] = LT^{-1}$, $[\epsilon] = L^2 T^{-3}$ and $[r] = L$, an expression for ϵ follows from dimensional analysis:

$$\epsilon \sim \frac{(\delta_r u)^3}{r}. \tag{2.7}$$

The Reynolds number associated to the scale r reads:

$$Re(r) \sim \frac{(\delta_r u) r}{\nu} \sim \frac{\epsilon^{1/3} r^{4/3}}{\nu}, \tag{2.8}$$

which yields an expression for the dissipative scale corresponding to $Re(l_d) = 1$:

$$l_d = \nu^{3/4} \epsilon^{-1/4} \tag{2.9}$$

Note that it is possible to derive a formal expression for S_3. Multiplying the Navier-Stokes equations (2.1) for $\boldsymbol{u}(\boldsymbol{x})$ by $\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r})$ yields an equation for the time evolution of

S_2 as a function of S_3, called *Karman-Howarth*, or simply KH relation. It may then be solved for S_3 in the statistically stationary state (for details of the calculations, see Frisch (1995); Landau & Lifshitz (1987)), yielding the only exact result in turbulence known as the *4/5-law* (Kolmogorov, 1941):

$$S_3(r) = -\frac{4}{5}\epsilon r. \tag{2.10}$$

This relation implies that there is a constant energy flux in the inertial interval of scales, equal to the one injected at the stirring scale. It also displays that the velocity increments are *negatively skewed*: particles getting closer are more probable than particles separating.

Kolmogorov then assumed strict self-similarity for the velocity differences, i.e. the existence of a unique exponent h such that $\delta^{\|}_{\lambda r}\boldsymbol{u} \sim \lambda^h \delta^{\|}_{r}\boldsymbol{u}$. This implies that $\zeta(q) = hq$ is a linear function of q. Requiring (2.10), one gets $h = 1/3$ hence:

$$S_q(r) = C_q(\epsilon r)^{q/3}, \tag{2.11}$$

where the C_p's are universal dimensionless constants. Furthermore,

The quantity S_2 is actually linked to the power-spectra of the velocity. Let us first define the energy density $e(\boldsymbol{k})$ in the Fourier space (we note $\mathcal{F}$ the Fourier transform), with $\boldsymbol{k}$ the wave-vector:

$$e(\boldsymbol{k}) = \frac{1}{(2\pi)^d}\frac{\|\hat{\boldsymbol{u}}(\boldsymbol{k})\|^2}{2V} = \mathcal{F}\left[\sum_i R_{ii}(\boldsymbol{r})\right]. \tag{2.12}$$

$V = L^d$ is the domain volume and $R_{ij}(\boldsymbol{r}) = \langle u_i(\boldsymbol{x})u_j(\boldsymbol{x}+\boldsymbol{r})\rangle$ is the velocity correlation tensor. The following convention for the Fourier transform is used: $\hat{\boldsymbol{u}}(\boldsymbol{k}) = \int_{\mathbb{R}^d} e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}$. The kinetic energy spectrum may then be defined in various forms, such as the two following:

$$E(k) = \int_{\mathbb{R}^d} \mathrm{d}\boldsymbol{k}'\delta(\|\boldsymbol{k}'\| - k)e(\boldsymbol{k}') = \int_{\Theta_k} \mathrm{d}\Omega_k k^{d-1}\frac{\|\hat{\boldsymbol{u}}(\boldsymbol{k})\|^2}{2}, \tag{2.13}$$

with Θ_k the hypersphere in Fourier space of radius k and Ω_k the solid angle element. Owing to the Parseval theorem, the mean energy in our system can be evaluated either in Fourier or physical space via:

$$E = \frac{1}{V}\int_{\mathbb{R}^d}\frac{\|\boldsymbol{u}(\boldsymbol{x})\|^2}{2}\,\mathrm{d}\boldsymbol{x} = \int_0^\infty E(k)\,\mathrm{d}k. \tag{2.14}$$

The following relation gives the equivalence between structure functions and spectra. Given a *power-law* spectrum:

$$F(k) \propto k^{-n}, \qquad 1 < n < 3, \tag{2.15}$$

then the second order structure function is also a power-law with (Frisch, 1995):

$$\left\langle (f(x) - f(x+r))^2 \right\rangle \propto |r|^{n-1}. \tag{2.16}$$

The energy spectrum is sometimes more easily interpretable than the structure function which is its physical pendant. It is also easily obtainable in numerical simulations that use spectral methods to integrate the velocity fields (see appendix A.1).

Using (2.11) and (2.16) for $q = 2$, one gets

$$E(k) = \epsilon^{2/3} k^{-5/3}, \tag{2.17}$$

which is also the only dimensionally correct combination of ϵ and k. This spectrum shape has been effectively observed in numerical and experimental works. However, in three dimensions, self-similar hypothesis shows to be more and more inexact as the moment q increases (Frisch, 1995). Indeed, the exponent $\zeta(q) = \frac{\mathrm{d}\log S_q}{\mathrm{d}\log r}$ has been shown to be a strictly *concave* function of q, rather than linear. This is called *anomalous scaling*.

2.2 Turbulence in two dimensions

This thesis work mainly involves two-dimensional turbulent flows, which display some interesting features that are phenomenologically different from their three-dimensional pendants.

2.2.1 Two dimensional Navier–Stokes equations

In incompressible flows, $\partial_i u_i = 0$ so that the two-dimensional velocity field is fully determined from the function $\psi(\boldsymbol{x}, \boldsymbol{y}, t)$ via the relation $\boldsymbol{u} = \nabla^{\perp}\psi = (\partial_y u_x, -\partial_x u_y)$ (the sign of $\nabla^{\perp}$ may vary in the literature). The level sets of ψ represent the stream-lines, with $\boldsymbol{u}$ being everywhere tangent to these level curves, hence the name stream-function for ψ. Vorticity is related to ψ by the relation $\omega = \nabla \times \boldsymbol{u} = -\nabla^2\psi$. The evolution equation for ψ reads:

$$\partial_t \psi + \frac{1}{\nabla^2}\{\psi, \nabla^2\psi\} = \nu\nabla^2\psi + f'_\psi \tag{2.18}$$

where $\{\}$ denotes the Poisson bracket, or Jacobian, such that $\{f, g\} = \partial_x f \partial_y g - \partial_x f \partial_y g$. Kraichnan (1967) already conjectured that energy would accumulate in the gravest mode k_{min} allowed by the boundary conditions (see below section 2.2.2). This accumulation of energy at large scale would eventually a condensate (illustrated in Figure 2.1), analogous to a Bose-Einstein condensate.

In the real world, mechanisms arise from various physical origins to prevent this large scale energy piling up, like Rayleigh friction in stratified fluids or the friction induced by the surrounding air in soap-film experiment. An additional linear term $-\alpha\psi$ is often added in Navier-Stokes equations to represent this friction, so that $f'_\psi = f_\psi - \alpha\psi$. α is thus the Ekman friction coefficient responsible for the large scale energy dissipation. It is often used in numerical simulations to reach a statistically stationary state, especially in the two-dimensional inverse cascade (Salmon, 1998). The origin of its linear form may

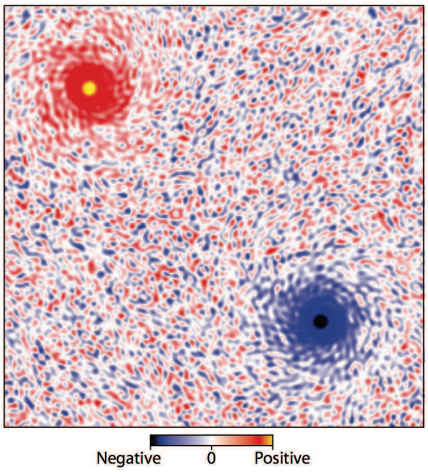


Figure 2.1: Vorticity condensate in the two dimensional inverse cascade. From Boffetta & Ecke (2012).

be exemplified the following way: near solid boundaries with no-slip condition, a laminar Poiseuille profile is usually admitted. Say the surface is horizontal at $z = h$, with z the vertical direction, then the velocity reads $\boldsymbol{u}(\boldsymbol{z}) = \frac{\alpha}{2\nu}(\boldsymbol{z} - h)^2$ and $\nu(\partial_x^2 + \partial_y^2 + \partial_z^2)\nabla^2\boldsymbol{u} \sim \nu(\partial_x^2 + \partial_y^2)\boldsymbol{u} - \alpha\boldsymbol{u}$. In experiments, it is also this drag form that is adopted for liquid friction on a soap film, or with the bottom of a container. Also ion-neutral collisions in ionospheric plasma give rise to such a friction.

The equivalent equation for the vorticity is obtained by taking minus the Laplacian of equation (2.18):

$$\partial_t\omega + \boldsymbol{u} \cdot \boldsymbol{\nabla}\omega = \nu\nabla^2\omega - \alpha\omega + f_\omega \tag{2.19}$$

ω is written in non-bold font to explicit that it is treated as a scalar quantity in the 2D plane: the vortex lines are always perpendicular to the flow plane. Equation 2.19, except for the forcing f_ω, expresses vorticity transport by the flow and dissipation by small-scale molecular viscosity and large-scale friction. One important difference between (2.19) compared to its 3D equivalent is the lack of the *vorticity stretching* term $(\boldsymbol{\omega} \cdot \boldsymbol{\nabla})\boldsymbol{u}$ which is identically 0 in 2D. In 3D, it is responsible for vorticity *amplification* in the vortex stretching direction due to angular momentum conservation.

2.2.2 The double cascade framework

Global invariants

Neglecting the large-scale friction and force terms in (2.19) and multiplying (2.19) by ω and averaging, one gets $D_t Z = \nu \nabla^2 \omega$ where

$$Z = \frac{1}{2} \left\langle \omega^2 \right\rangle \tag{2.20}$$

is the *enstrophy*. Thus, for inviscid flows ($\nu = 0$), enstrophy is conserved along Lagrangian trajectories[1]. Actually, $\int \omega^n \mathrm{d}^2\boldsymbol{x}$ is a constant of motion for all $n > 0$. In two-dimensional turbulence, enstrophy and energy are the quadratic invariants.

To give an intuition about the mechanism at play, the unforced Navier–Stokes equations for the velocity Fourier coefficients are first introduced:

$$\partial_t \hat{u}_i(\boldsymbol{k}) + \left(\delta_{ij} - \frac{k_i k_j}{k^2} \right) \sum_{\boldsymbol{k}=\boldsymbol{p}+\boldsymbol{q}} i p_j \hat{u}_i(\boldsymbol{p}) \hat{u}_j(\boldsymbol{q}) = -\nu \left\| \boldsymbol{k} \right\|^2 \hat{u}_i(\boldsymbol{k}). \tag{2.21}$$

The second term on the left-hand side is the Fourier transform of the non-linear term. It shows that the mode $\boldsymbol{k}$ interacts with modes $\boldsymbol{p}$ and $\boldsymbol{q}$ such that $\boldsymbol{p} = \boldsymbol{q} + \boldsymbol{k}$, i.e. $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{k}$ must form a triangle. This is called a *triadic interaction*, and conserves both energy and enstrophy.

In order to grasp the essence of energy transfers in two-dimensional turbulence, one can refer to the paper by Kraichnan (1967) who deduced the direction of the cascades of energy and enstrophy with statistical mechanics arguments. The term *cascade* refers to the fact that the injected energy (or enstrophy) is transferred at a *constant rate* through the scales. This transfer results from the non-linear term in (2.1) and takes place until molecular dissipation counter-balances at the dissipation scale.

Another phenomenological argument was already advanced by Fjørtoft (1953) to predict the direction (in the scale space) of these transfers. A triadic interaction between wavenumbers $k_1 < k_2 < k_3$ conserving both energy and enstrophy, we have that their variations must vanish: $\sum_i \Delta E_i = \sum_i k_i^2 \Delta E_i = 0$. Therefore:

$$\Delta E_1 = \frac{k_2^2 - k_3^2}{k_3^2 - k_1^2} \Delta E_2 \qquad \Delta E_3 = \frac{k_1^2 - k_2^2}{k_3^2 - k_1^2} \Delta E_2. \tag{2.22}$$

This implies that when mode k_2 loses energy ($\Delta E_2 < 0$), more energy will go into k_1 than k_3, indicating a direction of energy toward large scales. Similarly, more enstrophy must go into k_3 than k_1, so that enstrophy is transferred to small scales.

It also implies that the energy and enstrophy transfers must take place in two different directions in the $|\boldsymbol{k}|$ space.

---

[1] In three dimensions, an additional term appearing in the vorticity equation, called *vortex stretching*, violates this conservation.

The direct enstrophy cascade is considered to be the process responsible for vorticity filaments stretching and folding, creating stronger and stronger vorticity gradients until they are eventually dissipated by molecular viscosity. See (Kraichnan & Montgomery, 1980; Monin & Ozmidov, 1985) for details.

Spectrum scaling in the dual cascade

The energy and enstrophy spectra are related through $Z(k) = k^2E(k)$. In the inverse cascade, under the assumption of Kolmogorov phenomenology, the energy spectrum exponent is identical to the one in three dimensions:

$$E(k) = C_1\epsilon^{2/3}k^{-5/3}, \tag{2.23}$$

$$Z(k) = C_1\epsilon^{2/3}k^{-1/3}, \tag{2.24}$$

where ϵ is the energy dissipation rate and C_1 is a constant, determined to be in the range $\sim 5.8 - 7.0$ (Paret & Tabeling, 1997).

In the direct enstrophy cascade, the spectra are:

$$E(k) = C_2\epsilon_\omega^{2/3}k^{-3}[\ln(k/k_f)]^{-1/3}, \tag{2.25}$$

$$Z(k) = C_2\epsilon_\omega^{2/3}k^{-1}[\ln(k/k_f)]^{-1/3}. \tag{2.26}$$

where ϵ_ω represents this time the enstrophy dissipation rate, analogous to ϵ for energy. $k_f = 2\pi/l_f$ denotes the forcing wave-number corresponding to the forcing length l_f.

The presence of the logarithmic factor in (2.25) and (2.26) is a correction that ensures that the enstrophy flux is constant across the inertial range (see Kraichnan (1971); Rose & Sulem (1978) for details), This factor is important for regularity reasons. Indeed, the total enstrophy $Z = \int k^2E(k)\mathrm{d}k \sim \|\nabla \cdot \boldsymbol{u}\|^2$ with $E(k) \propto k^{-3}$ logarithmically diverges, and the velocity field is not differentiable. It also implies that enstrophy fluxes are less local (i.e., contributions to the flux at k can come from a much wider range of wave-numbers around k) than their energy pendants.

Those scalings for energy and enstrophy spectra have been indeed observed in numerical studies, already in Borue (1994), and experimentally in large-scale geophysical or quasi-2D stratified flows and soap films (Boer *et al.*, 1984; Rivera & Wu, 2000; Daniel & Rutgers, 2002). Figure 2.2, coming from a direct numerical simulation at very high resolution (Boffetta & Musacchio, 2010), illustrates these two regimes.

In the inverse cascade, one can also derive an expression for $S_3(r)$ in the same way than in three dimensions (see section 2.1). It reads:

$$S_3(r) = \frac{4}{3}\epsilon r. \tag{2.27}$$

Compared to (2.10), S_3 is positively skewed in two dimensions. In the direct cascade, the prediction is $S_3(r) = \frac{1}{8}\epsilon_\omega r^3$.

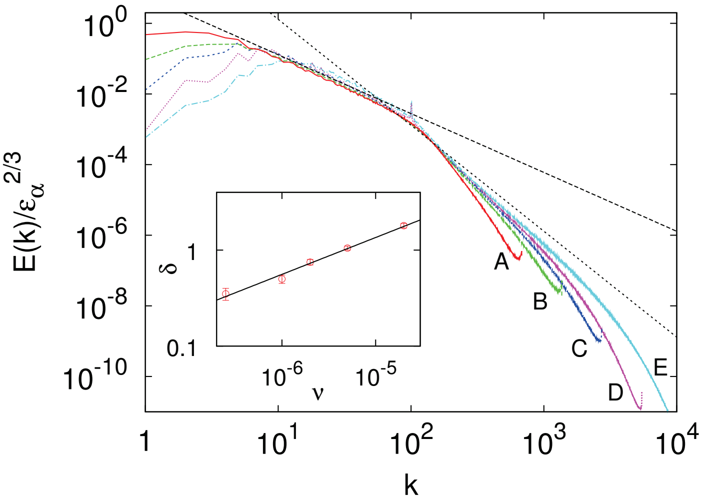


Figure 2.2: Energy spectra of two dimensional flows, obtained via Direct Numerical Simulations for spatial resolution up to 32768^2 (from Boffetta & Musacchio (2010)). In those simulations, the forcing wave-number is set at $k_f = 100$.

Effect of Ekman friction

Due to the energy flow toward the large scales in the two dimensional inverse cascade, it is necessary to provide a low-k energy sink if ones wants to achieve statistical stationary state. This is why the Ekman friction term is so important in two-dimensional simulations.

This term however has some impact on the flow structure. For example, it was shown by Nam *et al.* (2000); Bernard (2000); Boffetta *et al.* (2002) in the direct enstrophy cascade that as the friction coefficient α increases, so does the enstrophy spectrum slope, deviating more and more from the Kraichnan prediction, i.e. $E(k) \propto k^{-(3+\xi)}$ where ξ is related to the distribution of finite time Lyapunov exponents (see below). A consequence of (2.15) and (2.16) is that for $\alpha = 0$, $\xi = 0$ and for $0 < \xi < 2$, $\xi = \zeta_2^\omega$, where ζ^ω is the vorticity structure function exponent. The slope steeper than k^{-3} when $\alpha > 0$ implies the differentiability of the velocity field (see section 2.2.2) and the logarithmic correction in (2.25) and (2.26) is absent.

In the *inverse energy cascade* range, it may be shown that the effect is the reverse: as α decreases, energy builds up into a large scale condensate. The apparent effect is a steepening of its slope. This tendency is illustrated for various spectra coming from the simulations with resolution $N_x^2 = 4096^2$ performed in the framework of the study at chapter 3. The value of α was selected in such a way that the accumulation of energy is prevented at large scales, without depleting too much the inertial range cascade in order

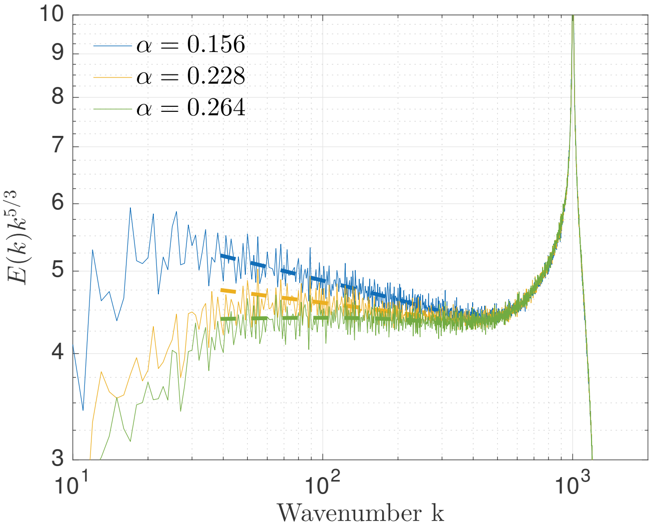


Figure 2.3: Energy spectra for various Ekman friction coefficients α in the two dimensional inverse energy cascade with $k_f = 10^3$. The spectra are compensated by $\|\boldsymbol{k}\|^{5/3}$. Dashed-lines represent fit in the inertial-range. Contrary to the direct cascade, the slope diminishes as the friction coefficient increases.

to get a spectral slope as close as possible to $k^{-5/3}$.

Intermittency in 2D turbulence

While the velocity field in three dimensions is known to be intermittent, the direct and inverse cascade deserve separate discussion in the two-dimensional case. In the inverse energy cascade, dimensional scaling was observed in numerical simulations by Boffetta *et al.* (2000) for the Lagrangian structure function of order up to $p = 7$, ruling out the possibility of intermittency similar to that in the 3D case. They nevertheless observed an antisymmetric part for high fluctuations of the longitudinal velocity differences, so that there is no Gaussianity. Other experimental and numerical works lead to the same conclusion (Paret & Tabeling, 1998; Chen *et al.*, 2006*b*; Xiao *et al.*, 2009).

In the direct cascade, velocity doesn't display any intermittency, so that its increment are Gaussian even at small scales. Rather, it is the vorticity structure function that displays anomalous scaling. The exponents $\zeta(p)$ of S_p may be related to the following dynamical argument. As stated in section 2.1, the Lyapunov exponent λ is obtained in the limit when two initially close trajectories in chaotic flows have diverged during an infinite time. When this time t is finite, these exponents depend on initial separations and are characterised, owing to the large deviation principle, by a probability density function $P(\lambda|t) = \sqrt{tG''(\lambda)/2\pi}\, e^{-tG(\lambda)}$ (Ott, 2002). The convex Kramer function $G(\lambda)$ is then related to the exponents by the relation $\zeta_{2n} = \min_h [2q, (G(\lambda) + 2q\alpha)/\lambda]$ (Neufeld *et al.*, 2000). This implies that the probability density function of the vorticity increment is not self-similar and deviates from Gaussian at small scales (Tsang *et al.*, 2005).

A third sign of intermittency is the multifractal property of the vorticity dissipation field, $\|\boldsymbol{\nabla}\omega(\boldsymbol{x})\|^2$. One way to get a measure of a chaotic attractor in a give phase space is to look at its Renyi dimension spectrum D_q. Dividing the phase space in (hyper)-cubes C_j of size ϵ, $\mathcal{D}_q$ is defined by (Renyi, 1970):

$$\mathcal{D}_q = \lim_{\epsilon \to 0} \frac{1}{1-q} \frac{\log\left(\sum_j \mu(C_j)^q\right)}{\ln(1/\epsilon)}. \tag{2.28}$$

μ is the natural measure associated with the attractor such that $\sum_j \mu(C_j) = 1$. $\mathcal{D}_q$ is a non-increasing function of q and is independent of q for non-fractal attractors. In particular, $\mathcal{D}_1$ is the information dimension and $\mathcal{D}_2$ the correlation dimension. These dimensions may be determined numerically using *box-counting* algorithms (see section 3.3.2 for an example of the measure of $\mathcal{D}_2$). Tsang *et al.* (2005) showed that anomalous scaling of the vorticity structure function yields multifractality of vorticity dissipation through the relation $D_q = 2 + \frac{\zeta_{2q} - q\zeta_2}{q-1}$.

2.2.3 Energy and enstrophy budgets

One can derive the equations for the evolution of the fluid energy and enstrophy by multiplying Navier-Stokes equations (2.18) respectively by the fluid velocity $\boldsymbol{u}$ and vorticity ω. Only the energy conservation terms are written, the case of enstrophy being analogous. The equation for the instantaneous variations of the shell-averaged energy content at wave numbers such that $\|\boldsymbol{k}\| = k$ reads:

$$\partial_t E(k) + \Pi(k) = -2\nu Z(k) + -\alpha E(k) + F(k). \tag{2.29}$$

$\Pi(k) = \langle \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \boldsymbol{\nabla u}) \rangle$ is the non-linear transfer contribution which satisfies

$$\int_0^\infty \Pi(k)\, \mathrm{d}k = 0. \tag{2.30}$$

This term also corresponds to the integral of the triadic interactions over the wave-numbers $\|\boldsymbol{k}\| = k$. This quantity, represented along the k axis, allows one to better visualise the cascades. Indeed, representing $\Pi^<(k) = \int_0^k \Pi(k)\, \mathrm{d}k$ on a plot with a logarithmic scale in the wavenumber dimension displays a plateau, i.e. a constant flux.

For example, in the inverse energy cascade, the energy goes from low to large values of k. $\Pi^<(k)$ is thus a source term for the large scales (larger than the forcing scale). Hence, as this term appears on the right hand side with a negative sign, it will give a negative

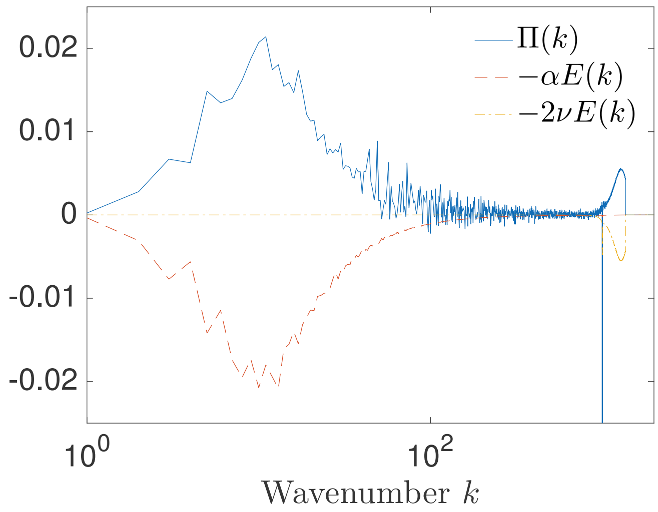


Figure 2.4: Illustration of the terms appearing in the spectral energy budget (2.29) in the inverse cascade at a resolution $N_x^2 = 4096^2$. The peak in the non-linear transfer is due to the stochastic forcing injected at the corresponding scale.

plateau. In the direct cascade, this term would be constant and positive for the enstrophy budget, as this quantity cascades to the small scales.

The energy dissipation has two contributions, which are on average negative for all k. One comes from molecular viscosity with total dissipation $\epsilon = \int_0^\infty 2\nu Z(k)\ \mathrm{d}k$, the other one from Ekman friction with total dissipation $\epsilon_\alpha = \int_0^\infty \alpha E(k)\ \mathrm{d}k$. The term $F(k) = \langle \hat{\boldsymbol{u}}(\boldsymbol{k}) \cdot \boldsymbol{f}(\boldsymbol{k}) \rangle$ denotes the input power at scales k, where the average is taken over wavenumbers with modulus k.

Figure 2.4 illustrates the non-linear term along with the friction term appearing in equation (2.29). Those simple relations may serve as a benchmark when designing Navier-Stokes solvers. In a statistically steady state, $\langle \partial_t E(k) \rangle$ is zero, and the other terms must, on average, balance.

2.3 Relative dispersion rates of Lagrangian trajectories

2.3.1 Lyapunov exponents

One important quantity to characterise the dynamics of transported elements in chaotic flows, coming back from the works of Lyapunoff (1907), describes how two tracers infinitesimally close do separate asymptotically in time following the continuous stretching, contractions and rotations of their separation vector. Tracers are particles that follow exactly fluid stream lines and can be thought as attached to fluid elements, hence their name (see 3.1.1). They are characterised by their position $\boldsymbol{X}$, and velocity $\boldsymbol{V} = \boldsymbol{u}(\boldsymbol{X})$, which is the fluid velocity at their position. Their equation of motion reads:

$$\dot{\boldsymbol{X}} = \boldsymbol{u}(\boldsymbol{X}, t). \tag{2.31}$$

Considering the tangent bundle in the phase-space $\boldsymbol{R}(t) = \delta\boldsymbol{X}(t)$ at point $\boldsymbol{X}(t)$, its evolution reads:

$$\frac{\mathrm{d}\boldsymbol{R}(t)}{\mathrm{d}t} = \sigma(t)\boldsymbol{R}(t), \tag{2.32}$$

with $\sigma_{ij} = \partial_j u_i(\boldsymbol{X}(t))$ denoting the Lagrangian strain matrix. The integral

$$J = \exp_0\left(\int_0^t \sigma(s)ds\right), \tag{2.33}$$

with $\exp_o$ the time-ordered exponential, is the Jacobian matrix such that $\boldsymbol{R}(t) = J\boldsymbol{R}(0)$.

Consider an initial volume of fluid which is evolved by the dynamic. It will be elongated along some directions and stretched along others. Actually, for each single trajectory and in the limit $t \to \infty$, the orientation of this ellipsoid's axis will have converged (due to the Multiplicative Ergodic Theorem by Oseledec (Oseledec, 1968)) in the directions of the eigenvectors $\boldsymbol{e}_j$ of the matrix $J^T J$. Indeed, with $\boldsymbol{R}(t) = \boldsymbol{J}\boldsymbol{R}_0$ then $\|R\|^2 = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{R}_0\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{R}_0$. $J^T J$ is a symmetric matrix, hence diagonalisable. Because it is also positive, its eigenvalues are positive and may written in the form of an exponential $e^{\lambda_j t}$, defining the *Lyapunov exponents*:

$$\lambda_j = \lim_{t\to\infty} \frac{1}{t} \ln\left(|J\boldsymbol{e}_j|\right) \quad j = 1, \ldots, 2d. \tag{2.34}$$

The evolution rate of a phase-space volume is given by $\theta = \sum_{j=1}^{2d} \lambda_j$. In incompressible flows, this sum is zero, and the volumes are thus conserved. To provide another example, as will be discussed in 4, for inertial particles, whose dynamic is *dissipative*, this sum is negative, yielding a contraction rate with a given characteristic time which is a property of the particles.

These Lyapunov exponents may be used to define a fractal dimension of a phase-space attractor, called the *Lyapunov dimension*. Ordering the exponents in decreasing order and defining the partial sum $S(i) = \sum_j^i \lambda_j$, then d_L is the interpolated index for which $S(d_L) = 0$. The Kaplan–Yorke conjecture (Kaplan & Yorke, 1979; Eckmann & Ruelle, 1985) then states that the information dimension of the attractor D_1 is equal to d_L.

2.3.2 Separation rates

The velocity of the relative separation between two tracers is not a trivial quantity in turbulence.

Denoting by $\boldsymbol{R}(t) = \boldsymbol{X}^i(t) - \boldsymbol{X}^j(t)$ the separation between two particles i and j at time t, the question is to know how fast such particles will migrate away from each other. It may depend on multiple factors like the initial and present separations $R(0), R(t)$ and the statistics of the fluid velocity differences projected on their separation vector

$$\delta_R u = (\|\boldsymbol{u}(\boldsymbol{x})\| - \|\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{R})\|) \cdot \frac{\boldsymbol{R}}{\|\boldsymbol{R}\|}. \tag{2.35}$$

Smooth flows

Consider first two particles separated by a distance below the dissipation scale l_d, i.e. in the *dissipative* range. The flow is then differentiable and Lipschitz in $\boldsymbol{R}$, i.e. $\delta_R \boldsymbol{u} \sim \sigma \boldsymbol{R}$, ensuring the unicity of the solution of (2.31). The particles then separate exponentially in time with the Lyapunov exponent λ:

$$\|\boldsymbol{R}(t)\| = \|\boldsymbol{R}_0\| \, e^{\lambda t}. \tag{2.36}$$

Non-smooth flows

Consider now two particles initially separated by $\|\boldsymbol{R}_0\|$ which is in the inertial sub-range. In this range, in the $3D$ direct cascade or $2D$ inverse cascade, velocity field is no longer smooth: $\langle \delta_r u \rangle \propto r^{1/3}$. This implies $dR^2/dt = 2\boldsymbol{R} \cdot \delta_R \boldsymbol{u} \propto R^{4/3}$ and:

$$\left\langle \|\boldsymbol{R}(t)\|^2 \right\rangle_{R_0} \propto g_R t^3, \tag{2.37}$$

where g_R is the Richardson constant and the average is taken over particle pairs initially at distance R_0. This rate is faster than diffusion ($R^2(t) \propto t$) and the regime is called *super-diffusive*, faster than what can be attributed to sole chaotic motions. Indeed, in the exponential separation (see paragraph above), the time for two particles to reach a scale R diverges logarithmically with R/R_0. On the contrary, the explosive separation does not depend on the initial separation R_0 and particles will always reach R in a finite time.

This observation was already predicted by Richardson (1926). He measured a scale-dependant diffusivity K which fits well with $K(r) \propto r^{4/3}$ on 4 decades. Indeed, a contaminant cloud of size r is only advected by vortices larger than r, and its diffusions results mainly from vortices of size r. From this result, he derived equation (2.37) using Fickian diffusion. This scaling law for $K(r)$ was later formulated in the framework of the Kolmogorov theory following the Obukhov hypothesis (Obukhov, 1941). If r is in the inertial range, the effective diffusivity $K(r)$ must only depend on r and ϵ, leading to $K(r) \propto \epsilon^{1/3} r^{4/3}$.

This super-diffusive behaviour however was showed by (Batchelor, 1950) to be preceded by a ballistic separation, i.e. with a velocity constant in time, for which we get $\left\langle \|\boldsymbol{R}(t)\|^2 \right\rangle = C(\epsilon R_0)^{2/3} t^2$. This regime is valid during the correlation time of the eddies of size R_0. This correlation time is typically of the order of the eddy turn-over time associated with the scale R, $\tau_R \propto \epsilon^{-1/3} R^{2/3}$. From this observation, Bourgoin (2015) and Thalabard *et al.*

(2014) successfully proposed to interpret the explosive separation as an iterative ballistic process.

This Richardson dispersion regime is also associated with another issue about the reversibility of pair separations: in three dimensions, the Richardson constant g_R is not the same when considering the *forward in time* evolution of pairs and its *backward in time* equivalent. Heuristically, this can be understood by the fact that the odd number of dimension allows a fluid ellipsoid to be elongated along more dimensions than those along which it can be squeezed. This would not be the case in two dimensions.

Part I

Turbulent dispersion and mixing

CHAPTER 3

Tracers dispersion in two dimensional turbulence

Contents

3.1 Introduction

Turbulent mixing is of particular concern in situations such as the formation of clouds through condensation of small water droplets (Grabowski & Wang, 2013), gas accretion in planet formation (Johansen *et al.*, 2007) or phytoplankton and nutrients distribution in the oceans (Mann & Lazier, 2013). Mixing refers to the evolution of an initial distribution of a scalar field (temperature, salinity, or the concentration of any substance...) by the fluid.

The mechanical stress induced by the stirred flow tends to deform the initial distribution of this field, and the multi-scale nature of turbulent flows gives rise to very complex shapes and patterns of the concentration field (Celani *et al.*, 2001). For example, while the scalar is also submitted to molecular diffusion, which tends to smooth out concentration gradients, mixing by stretching and compression in directions orthogonal to each other cause to reinforce these gradients by creating elongated concentration filaments.

Because incompressible flows preserve volumes, an homogeneous initial scalar concentration remains uniform at any later time. However, non-uniform initial patches of concentrations will be deformed by the swirling eddies and create locally high gradients. Figure 3.1 displays an instantaneous field of scalar concentration advected by a turbulent flow: large fronts and cliffs are seen along with rather uniform regions. These gradients form because turbulence brings close together trajectories of fluid elements carrying different scalar trajectories and history. Scalar differences over small scales grow in intensity while the front boundaries become thinner, until they are eventually dissipated by molecular viscosity. These large differences are responsible for strongly intermittent statistics in the scalar distribution (Sreenivasan & Antonia, 1997), i.e. the probability density function (pdf) of scalar value shows a departure from a Gaussian behaviour, and displays exponential tails (Pumir *et al.*, 1991) that result from rare, extreme events. They prove that a restrictive vision considering a large number of small-scales, uncorrelated stretching events for the scalar distribution, which would yield Gaussian pdf through the central-limit theorem, is not correct.

Interestingly, the passive scalar is strongly intermittent both in $2D$ and $3D$ even in the absence of intermittency in the velocity field itself in $2D$, and also in simple random Gaussian velocity fields (Shraiman & Siggia, 2000). Similarly to the velocity increments in 3D or the vorticity in the 2D direct cascade, the scaling exponents of the scalar structure function $S^n_\theta(r) = \langle(\theta(\boldsymbol{x}) - \theta(\boldsymbol{x}+\boldsymbol{r}))^n\rangle \propto r^{\zeta(n)}$ are not linear: $\zeta(n) \neq n/3$ (see section 2.1.1). Scale invariance is thus broken and S^n reads (Celani & Vergassola, 2001):

$$S^n_\theta(r) \propto r^{\zeta^{\mathrm{dim}}(n)} \left(\frac{L}{r}\right)^{\zeta^{\mathrm{dim}}(n)-\zeta(n)} \tag{3.1}$$

The difference $\zeta^{\mathrm{dim}}(n) - \zeta(n)$ is the correction for the anomalous scaling, and the broken scale invariance manifests in the presence of L although $r \ll L$.

Some analytical models for the velocity correlations, like the Kraichnan model (Kraichnan, 1994), allow to recover predictions about the behaviour of $\lim_{n\to\infty}\zeta(n)$. In the Kraichnan ensemble, the two-points, two-times velocity correlation are:

$$\langle v_i(\boldsymbol{x}_1,t_2)v_j(\boldsymbol{x}_2,t_2)\rangle = D_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)\delta(t_1-t_2) \tag{3.2}$$

with

$$D_{ij}(\boldsymbol{r}) = D_0\delta_{ij} - D_1 r^{\xi}[(d+\xi-1)\delta_{ij} - \xi\frac{r_i r_j}{r^2}] \tag{3.3}$$

for r smaller than the integral scale. ξ denotes the degree of roughness of the flow: the velocity field smoothness increases with ξ and is differentiable for $\xi = 2$.

In this model, and under the additional assumption of high dimensionality, $d \gg \zeta(2)$, it was analytically shown in Balkovsky & Lebedev (1998) that there exists a critical order n_c such that $\forall n > n_c$, $\zeta(n)$ is independent of n This asymptotic behaviour seems also to be observed with direct numerical simulations of two dimensional inverse cascade, where it was estimated in Celani *et al.* (2000).

As vorticity and passive scalar share the same transport equation, it is tempting to compare their scaling laws to see if the exhibit similarities. However, the direct link between ω and u make the equation for ω non-linear, which can lead to discrepancies for small scale quantities. For example, Dubos & Babiano (2003) have shown using numerical simulations that this difference is responsible for faster temporal fluctuations of the vorticity gradients. In Boffetta *et al.* (2002), a correspondence is made between the intermittency of vorticity and that of a passive scalar transported by the flow, showing that $\zeta_p^\omega = \zeta_p^\theta \; \forall p$. This correspondence may be explained using the following *ad-hoc* argument (Tsang *et al.*, 2005) based on the Lyapunov exponent λ (see section 2.3). Since $\lambda \sim \left\langle \|\nabla u\|^2 \right\rangle^{1/2} \sim \sqrt{\int_{k_f}^{\infty} k^2 E(k)\, \mathrm{d}k}$ and assuming, then $\lambda \sim k_f^{-\xi/2}$. Thus λ (and ∇u) characterising small separations stretching, are determined by large scale structures, and the small scale vorticity components behave like scalar advected by the large scale flow.

Structure functions of order n are linked to the equal time n-point correlation function of the scalar field. For example, consider the following equality for $n = 2$:

$$\begin{aligned} S_2(r,t) &= \left\langle \left(\theta(\boldsymbol{x}+\boldsymbol{r},t) - \theta(\boldsymbol{x},t)\right)^2 \right\rangle && (3.4)\\ &= \left\langle \theta(\boldsymbol{x})^2 \right\rangle + \left\langle \theta(\boldsymbol{x}+\boldsymbol{r})^2 \right\rangle - 2\left\langle \theta(\boldsymbol{x}+\boldsymbol{r},t)\theta(\boldsymbol{x},t) \right\rangle && (3.5)\\ &= 2\left(C_2(0,t) - C_2(r,t)\right). && (3.6) \end{aligned}$$

The last equality results from homogeneity and isotropy and $C_2(r,t) = \langle \theta(\boldsymbol{x}+\boldsymbol{r},t)\theta(\boldsymbol{x},t) \rangle$. Averages are taken over the positions $\boldsymbol{x}$.

The generalisation of this quantity to n-points displays the link with the n-point joint transition probability for the Lagrangian motion. For a set of n particles initially at position $x_0, \dots, x_n$ at instant t_0:

$$\begin{aligned} C_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n; t) &= \langle \theta(\boldsymbol{x}_1, t), \dots, \theta(\boldsymbol{x}_2, t) \rangle && (3.7)\\ &= \int_{t_0}^{t} \theta(\boldsymbol{x}_1^0, t), \dots, \theta(\boldsymbol{x}_2^0, t_0)\; p_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n, t | \boldsymbol{x}_1^0, \dots, \boldsymbol{x}_n^0, t_0)\; \mathrm{d}\boldsymbol{x}_1^0 \dots \mathrm{d}\boldsymbol{x}_n^0 && (3.8) \end{aligned}$$

where $p(\dots)$ expresses the joint probability that the n trajectories initially at positions $\boldsymbol{x}_1^0 \dots, \boldsymbol{x}_n^0$ are transported at $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ at time t.

The quantity C_n can be related to the joint motion of n particles. It has a geometrical interpretation in terms of Lagrangian trajectories. For example, in Celani & Vergassola (2001), the intermittency of the passive scalar advection is attributed to long lasting clustering of n-tuple of particles. In Bianchi *et al.* (2016), the shape of spherical puffs of particles emitted is monitored as a function of time, showing that although the puff is initially spherical, the quick and strong distortions prevent the cloud to return back to a spherical shape at later times. It is also shown not to affect much large scale transport statistics, like the pdf of durations of hits and between hits of a downstream target.

In particular, the two-point scalar correlation allows one to express pair dispersion statistics. This correspondence was for example used in Boffetta & Celani (2000) to link frequent pairs encounter and scalar fronts formation. This object, $C_2(r)$ may be analytically derived only under drastic constrain on the flow, like for example the Kraichnan ensemble, In such a flow, Celani *et al.* (2007) have studied scaling properties of a scalar continuously emitted from a point source and derived an exact relation for the two-points equal-time scalar correlation function $C(\boldsymbol{x}_1, \boldsymbol{x}_2, t) = \langle \theta(\boldsymbol{x}_1, t)\theta(\boldsymbol{x}_2, t)\rangle$, demonstrating the persistence of inhomogeneities at small scales.

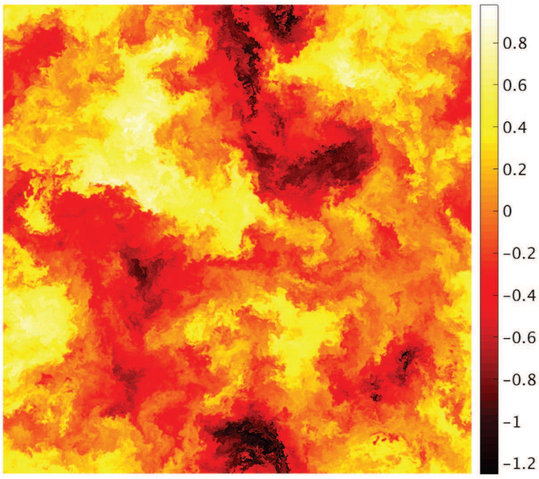


Figure 3.1: Illustration of a scalar field mixed by turbulent flow, representing a 2D slice from a 3D DNS simulation at $N_x^3 = 4096^3$ with a mean gradient scalar source. When initial inhomogeneities or inhomogeneous scalar sources are present, mixing by eddies create fronts where the scalar variations over very small scales are of the same order than the rms value itself.

3.1.1 Diffusion at long times

In this section, the correspondence between long time displacements of discrete particles and the scalar diffusive behaviour is explicited.

Tracers are particles solely advected by the flow, and thus that perfectly follow streamlines. Denoting their position by $\boldsymbol{X}$ and velocity $\boldsymbol{V}$, their equation of motion simply reads:

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{V}(t), \tag{3.9}$$

$$\dot{\boldsymbol{V}}(t) = \boldsymbol{u}(\boldsymbol{X}, t), \tag{3.10}$$

where $\boldsymbol{u}(\boldsymbol{X}, t)$ is the flow velocity at time t and at the particle position $\boldsymbol{X}$.

Starting from position $\boldsymbol{X}(0)$ at $t = t_0$, defining its *displacement* by :

$$\boldsymbol{X}(t) - \boldsymbol{X}(0) = \int_{t_0}^{t} \boldsymbol{u}(\boldsymbol{X}(\tau), \tau)\, \mathrm{d}\tau. \tag{3.11}$$

The mean square displacement then reads:

$$\left\langle \|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2 \right\rangle = \int_0^t \int_0^t \left\langle \boldsymbol{u}(\boldsymbol{X}(\tau), \tau) \cdot \boldsymbol{u}(\boldsymbol{X}(\tau'), \tau') \right\rangle\, \mathrm{d}\tau\, \mathrm{d}\tau'. \tag{3.12}$$

In turbulent flows, the exact behaviour of the integrand is of course unknown. But there are two asymptotics in which we can approximate this integral.

The first one, trivial, is when considering very small durations $t - t_0$, such that $\boldsymbol{u}$ may be considered constant, or *self-correlated* during this time interval, leading to $\boldsymbol{X}(t) = \boldsymbol{X}_0 + \boldsymbol{u}(t_0)(t - t_0)$, so that

$$\left\langle \|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2 \right\rangle = u_{\mathrm{rms}}^2 (t - t_0)^2. \tag{3.13}$$

where $u_{\mathrm{rms}}^2 = \left\langle u_x^2 + u_y^2 \right\rangle$ is the mean square velocity averaged over the two dimensions x and y.

The second exploits the fact that the autocorrelation of the tracers velocity $\boldsymbol{u}$ decreases to 0 over a finite time scale T_{C_L}, the *Lagrangian correlation time*, defined as:

$$T_{C_L} = \frac{1}{u_{\mathrm{rms}}^2} \sum_i \int_0^{\infty} \langle u_i(\boldsymbol{X}(0), 0) u_i(\boldsymbol{X}(\tau), \tau) \rangle\, \mathrm{d}\tau. \tag{3.14}$$

In the limit of large times ($t \gg T_{C_L}$), (3.12) becomes:

$$\left\langle \|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2 \right\rangle \sim \int_0^t T_{C_L} u_{\mathrm{rms}}^2\, \mathrm{d}\tau \sim T_{C_L} u_{\mathrm{rms}}^2 t. \tag{3.15}$$

This result may also be formulated the following way: decomposing the interval $[t_0, t]$ into a subset of N intervals of length T_{C_L}, $[t0 + iT_{C_L}, t0 + (i+1)T_{C_L}]$.

$$\boldsymbol{X}(t) - \boldsymbol{X}(0) = \sum_{i=0}^{N} \int_{t_0 + i T_{C_L}}^{t_0 + (i+1) T_{C_L}} \boldsymbol{u}(\tau)\, \mathrm{d}\tau \tag{3.16}$$

This long-time displacement is thus a sum of random variables identically distributed, hence, invoking the central limit theorem, it is itself a random Gaussian variable with variance:

$$\left\langle \|\boldsymbol{X} - \boldsymbol{X}(0)\|^2 \right\rangle = 2u_{\mathrm{rms}}^2 T_{C_L} t. \tag{3.17}$$

Beside the Lagrangian description of particle dispersion, a continuous concentration field $\theta(\boldsymbol{x})$ can be defined, which is also transported along the fluid trajectories.

Let us define this scalar quantity in terms of emitted tracers from a source situated at position x_S which emits particles at a constant rate $\phi_S(\tau) = \phi_S$ starting from the instant t_0. The expression of $n^*(\boldsymbol{x}, t)$ counting the number of particles reaching $\boldsymbol{x}$ at time t then reads:

$$n^*(\boldsymbol{x}, t) = \int_{t_0}^{t} \phi_S \, \delta(\boldsymbol{x}(\tau; \boldsymbol{x}, t) - \boldsymbol{x}_S) \, \mathrm{d}\tau \tag{3.18}$$

where $\boldsymbol{x}(\tau; \boldsymbol{x}, t)$ is the position of the particle at time τ given that it is at position $\boldsymbol{x}$ at time t. ϕ_S is an arbitrary number of particles injected at $\boldsymbol{x}_S$ at time τ.

The *ensemble* average of n^*, i.e. over a large number of realisations, defines a number density field:

$$n(\boldsymbol{x}, t) = \langle n^*(\boldsymbol{x}, t) \rangle = \int_{t_0}^{t} \phi_S \, p(\boldsymbol{x}_S, \tau | \boldsymbol{x}, t) \, \mathrm{d}\tau \tag{3.19}$$

where $p(\boldsymbol{x}_S, \tau | \boldsymbol{x}, t)$ is the transition probability to travel from $\boldsymbol{x}_S$ at time τ to $\boldsymbol{x}$ at time t. Notice that with such a definition, n^* and n both have units of number of particles per unit area, or L^{-2}. This is a consequence of the function $\delta(\boldsymbol{x})$ having units of L^{-2}. In the following, the field $n(\boldsymbol{x}, t)$ is sometimes used in place of the scalar concentration $\theta(\boldsymbol{x}, t)$. θ having units of a mass density, the two are simply related by $\theta(\boldsymbol{x}) = m_p n(\boldsymbol{x})$ with m_p the constant mass of one particle.

As a remark, let us stress that formally, the following distinction has to be kept in mind. The scalar quantity $\theta(t, \boldsymbol{x})$ is a transported quantity along a Lagrangian trajectory between $\boldsymbol{x}_0$ at t_0 and $\boldsymbol{x}$ at time t:

$$\theta(\boldsymbol{x}, t) = \theta(t_0, \boldsymbol{X}(t_0; t, \boldsymbol{x})) = \int \delta(\boldsymbol{x}_0 - \boldsymbol{x}(t_0; t, \boldsymbol{x})) \theta(t_0, \boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0, \tag{3.20}$$

while the field $n(\boldsymbol{x}, t)$ results from the forward Lagrangian flow:

$$n(\boldsymbol{x}, t) = \int \delta(\boldsymbol{x} - \boldsymbol{X}(t; t_0, \boldsymbol{x}_0)) n(t_0, \boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0. \tag{3.21}$$

Equation (3.17) describes a diffusive process for the particle displacements. The link between θ and Lagrangian trajectories (3.20) suggests that the scalar obeys at long times the differential equation:

$$\partial_t \theta = \kappa_T \nabla^2 \theta + S(\boldsymbol{x}), \tag{3.22}$$

which is the scalar pendent of equation (3.17). The operator $\kappa_T \nabla^2$ stands for the turbulent diffusion with $\kappa_T = T_L u_{\mathrm{rms}}^2$. $S(\boldsymbol{x})$ is the source term. In the case we considered, it is a punctual source of constant emission rate ϕ_S at position $\boldsymbol{x}_S$. Hence $S(\boldsymbol{x}) = \phi_S \delta(\boldsymbol{x} - \boldsymbol{x}_S)$.

Once again, equations (3.22) and (3.17) are only valid when considering long times $t - t_0$ since emission or long distances R from the source, i.e. $R > u_{\mathrm{rms}} T_{C_L}$, where T_L is the time during which Lagrangian velocities are correlated, or equivalently $(t - t_0) > R/u_{\mathrm{rms}}$.

The advection-diffusion of the passive scalar can be obtained by expressing the conservation of θ in an elementary volume and reads:

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \nabla \theta = \kappa \nabla^2 \theta, \tag{3.23}$$

with κ the molecular diffusion coefficient and $\boldsymbol{u}$ the convection velocity. The link between (3.23) and (3.22) deserves a comment. Decomposing u and θ into a mean and fluctuating part, i.e. $\theta(\boldsymbol{x}, t) = \overline{\theta(\boldsymbol{x})} + \theta(\boldsymbol{x}, t)'$ and $\boldsymbol{u} = \overline{\boldsymbol{u}(\boldsymbol{x})} + \boldsymbol{u}(\boldsymbol{x}, t)'$, we may then substitute in (3.23) and taking the average. Given that $\langle \boldsymbol{u}' \rangle = 0$ and $\langle \theta' \rangle = 0$, one obtains:

$$\partial_t \overline{\theta} + \overline{\boldsymbol{u}} \cdot \nabla \overline{\theta} = \kappa \nabla^2 \overline{\theta} - \boldsymbol{\nabla} \cdot \overline{\boldsymbol{u}' \theta'}. \tag{3.24}$$

The last term on the right-hand side denotes the turbulent mixing. To close this equation, the approximation $-\overline{\boldsymbol{u}' \theta'} = \kappa_T \nabla \theta$ is made, with $\kappa_T > \kappa$, yielding (3.22) (omitting the source). As an example, consider atmospheric dispersion. Molecular diffusivity of carbon dioxide in the air is $\sim 16 \times 10^{-6} m^2 s^{-1}$ while turbulent diffusion is often estimated as $\sim 30 m^2/s$, hence several orders of magnitude greater.

The Green function with $\boldsymbol{u} = \boldsymbol{0}$ of (3.23) is a d-dimensional Gaussian whose variance decreases linearly with time:

$$\theta(\boldsymbol{x}, t) = \frac{M(t)}{(2\pi |\boldsymbol{\Sigma}(t)|)^{d/2}} \exp\left(-\frac{1}{2} \boldsymbol{x}^T \boldsymbol{\Sigma}(t)^{-1} \boldsymbol{x} \right). \tag{3.25}$$

$M(t) = \int \theta(\boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{x}$ is the total mass and is a quantity constant in time, hence preserved by diffusion. $\boldsymbol{\Sigma}$ is the covariance tensor. Note that the solution for a diffusion process in a medium moving with constant velocity $\boldsymbol{U}$ is straightforwardly obtained by applying the Galilean transformation $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{U} t$.

If the diffusion is independent along each direction, the tensor $\boldsymbol{\Sigma}$ is diagonal. Furthermore, if the diffusion is isotropic, the solution reads

$$\theta(r, t) = \frac{A}{(2\pi \sigma(t)^2)^{d/2}} \exp\left(-\frac{r^2}{2\sigma(t)^2} \right) \tag{3.26}$$

and only depends on time and distance r from the mean.

3.1.2 Continuous source

Consider now a continuous emission of particles that takes place during a duration t. Due to the linearity of equation (3.23), the solution can be expressed as the superposition of an infinite number of emitting sources from t' to $t' + dt$, each emitting a quantity $\phi_S dt$. The solution then reads:

$$\theta(\boldsymbol{x}, t) = \frac{1}{(2\pi)^{d/2}} \int_0^t \frac{1}{(t-t')^{d/2}} \exp\left(-\frac{r^2}{4\kappa(t-t')}\right) dt' \tag{3.27}$$

The value of this integral converges on $[0, \infty]$ only for $d > 2$. For $d = 2$, it diverges logarithmically. Why is that so? Actually, diffusion at long times translates into Wiener process trajectories followed by the particles. And it is known that the Wiener process is a recurrent stochastic process in dimensions less or equal than two (Van Kampen, 1992).

3.2 Concentration and mass fluctuations of particles emitted by a continuous point source

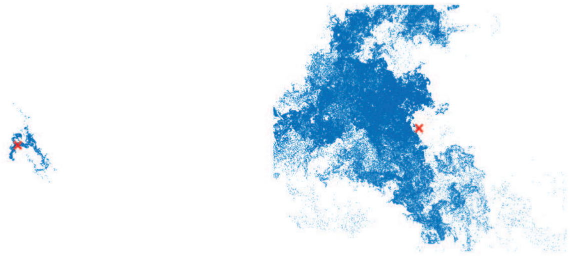
The large scalar temporal correlations play a role in what is called *vortex trapping*. Indeed, particles trapped inside a long-lived coherent vortex allows for interactions which take place at a distance of the size of the eddy to last longer and affect both the suspended solid phase as well as the carrier flow. This phenomenon is for example currently believed to play a key role in planetesimal accretions. Indeed, in planet formation, one challenging step is the understanding of the formation of planetesimals of the kilometre size, and the existence of long-living vortices in protoplanetary disks capable to concentrate large dust concentrations constitute a promising theory (Meheut *et al.*, 2012).

The challenge of the system that is studied in this chapter, i.e. the continuous emission from a source, lies in the fact that both spatial and temporal correlations play a role in the n-point concentrations. The addition of the time difference in the correlations adds a non trivial complexity. In this chapter, the problem of continuous mass release is addressed.

Having introduced to the reader the dispersion dynamics of both Lagrangian tracers and scalar, the chosen framework to study the dispersion from a continuous point source in two-dimensional turbulence is now described.

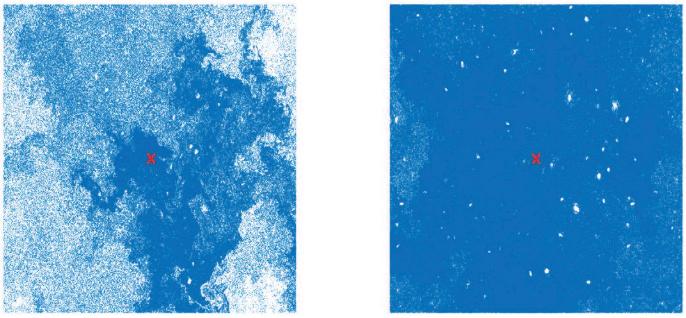
3.2.1 Fluid phase integration

The flow regime we considered is the inverse turbulent cascade. Direct numerical simulations have been performed using pseudo-spectral (Fourier) scheme in a d-dimensional periodic domain. The flow is forced at high wave-numbers. To maximize the inertial range and to minimise the range of scales affected by viscosity, we chose to implement hyperviscosity, which translates into a higher power of the Laplacian $p > 2$ in the Navier-Stokes



(a) $t = 0.22\, T_L$

(b) $t = 1.12\, T_L$



(c) $t = 4.5\, T_L$

(d) $t = 33.7\, T_L$

Figure 3.2: Illustration of recurrence phenomenon. In a few integral times T_L, particle distribution becomes nearly uniform in the domain due to large amount of returns near the source. The cut-off distance is $R_{\max} = 2L$ and width of the window is L.

equations. In all the simulations presented here, we have chosen $p = 8$. An illustration of the kinetic energy spectrum for a resolution $N_x^2 = 4096^2$ is displayed in Figure 3.3.

In the two-dimensional inverse cascade, this number is defined as the ratio between the friction and forcing scales:

$$Re_\alpha = \left(\frac{l_\alpha}{l_f}\right)^{2/3} = \left(\frac{k_f}{k_\alpha}\right)^{2/3}, \tag{3.28}$$

where l_α is the friction scale defined by $l_\alpha = \epsilon^{1/2}\alpha^{-3/2}$, and ϵ is the intensity of the nonlinear energy flux going from small to large scales.

More details about the numerical integration of two-dimensional turbulence may be found in appendix A.1.

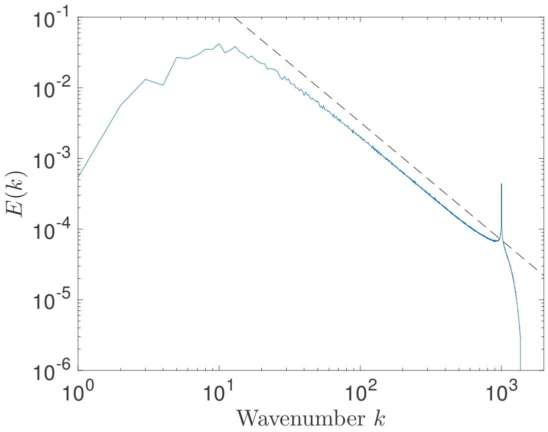


Figure 3.3: Fluid kinetic energy spectrum for the two-dimensional inverse cascade with hyper-viscosity. The resolution is $N_x^2 = 4096^2$ and the flow is forced at $k = 1000$ with a white noise. The black dashed line corresponds to the dimensional scaling $k^{-5/3}$.

3.2.2 Injection mechanism

Numerically, a number of particles ϕ_S is injected into the domain for a period $t = 1$. The symbol ϕ_S thus stands for the injection rate, in units of number of particle per unit time $[T]^{-1}$, or equivalently mass per unit time, $[M]/[T]$, through the relation $m = \rho_p a^3 \phi_S$, with $\rho_p = \rho_f$ the particle density and a the particle radius, considered as infinitesimal. During a time step Δt, $\phi_S \Delta t$ particles are thus emitted.

Two different mechanisms were considered for the injection of particles from the source. The first one consists of a uniform seeding around the source, either in a square region with sides R_x and R_y, or a circular region of radius R. This may be used to emit regular puffs of particles at a given period T_{inj} and is used for example in Scatamacchia *et al.* (2012) to study strong deviations from Richardson separation law, or in Celani *et al.* (2014) to determine the probabilities to detect concentrations above a given threshold downstream of the emission.

Another way of releasing particles into the domain is proposed, the reason why will be explained in section 3.3.3. Initially, at the beginning of the simulation, a first particle is randomly seeded close to the source. Its position and identifier is noted $\boldsymbol{X}_0$. Then, $\phi_S\,\Delta t$ particles are injected at positions which are linearly interpolated between the source and $\boldsymbol{X}_0$:

$$\boldsymbol{X}_i = \boldsymbol{X}_0 + i\frac{(\boldsymbol{x}_S - \boldsymbol{X}_0)}{\phi_S \Delta t} \qquad i = 1, \ldots, \phi_S \Delta t. \tag{3.29}$$

During the next time step, all particles will have moved further, and $\boldsymbol{X}_0$ takes the value of the position of the last emitted one which becomes the new reference for the next time step. This mechanism, referred later as line injection, requires the period T_{inj} to be short enough so that $\|\boldsymbol{X}_0 - \boldsymbol{x}_S\|$ is not too large compared to small scale structures, otherwise the injection creates unrealistically aligned structures traversing (and not affected by) eddies. Hence, it should be at most comparable to the dissipation time scale. In our case, the chosen period is equal to the time step ($T_{\text{inj}} = \Delta t$) which is the best possible choice to respect the approximation of a continuous source.

3.2.3 Removal mechanism

Boundary conditions must be implemented to remove particles from the system in order to reach a statistical steady state for the total number of particles. Furthermore, if one wants to achieve reasonable statistics to measure, for example, fractal dimensions at a given distance R from the source, both a sufficient number of particles need to be present and during a time long enough to be able to perform ensemble averages as large as possible. This may quickly leads to unmanageable number of particles present in the domain and pauses difficulties for both computation capacity and memory. Hence, *absorbing boundary condition* are used: as soon as a particle goes beyond a given criterion it is removed from the domain. In our experiments, depending on the quantity of interest, two different choices were implemented. The first one is a spatial criterion, allowing particles to live only inside a given boundary Γ. In all our simulations, this frontier consists of a circle centered around the source with a fixed radius R_{max}. The second considered boundary condition is a temporal one, allowing particles to live only until a maximum age $t_{p_{\text{max}}}$.

Additional details regarding the used software, *LAGSRC2D*, may be found in appendix A.2.

3.3 Results

3.3.1 One point dispersion

Mean square displacements and Lagrangian velocities autocorrelation

The mean square displacement of particles is defined as:

$$\left\langle R(\tau)^2 \right\rangle = \left\langle \|\boldsymbol{X}_p(t) - \boldsymbol{X}_p(t-\tau)\|^2 \right\rangle . \tag{3.30}$$

For small τ, as long as the Lagrangian velocity of the tracers is correlated, these are expected to migrate from their initial position at a constant speed, in average, yielding $\left\langle R(t)^2 \right\rangle = u_{\text{rms}}^2 t^2$. For times longer than the Lagrangian velocity correlation time T_{C_L}, the diffusive regime is expected to be recovered with $\left\langle R(t)^2 \right\rangle \propto t$.

T_{C_L} is defined in the following way. The Lagrangian velocity autocorrelation functions are given by:

$$C(\tau) = \frac{\langle \boldsymbol{V}_p(0) \cdot \boldsymbol{V}_p(\tau) \rangle}{\left\langle \|\boldsymbol{V}_p\|^2 \right\rangle}. \tag{3.31}$$

In statistically stationary flows, this quantity is independent of time and depends only on the time increment τ. It is well known that this quantity decays exponentially with time as it was already conjectured in Taylor (1921). From this function, the Lagrangian correlation time reads:

$$T_{C_L} = \int_0^\infty C(\tau)\, \mathrm{d}\tau. \tag{3.32}$$

When τ is small, the correlation $C(\tau)$ is constant and does not move appreciably away from unity. The velocity is differentiable with respect to τ and $C(\tau)$ may be developed in power series around $\tau = 0$. To the leading order, one has:

$$\left\langle \|\boldsymbol{X}_p(t) - \boldsymbol{X}_p(0)\|^2 \right\rangle = 2 \int\limits_0^t (t-\tau) C(\tau) \mathrm{d}\tau = u_{\mathrm{rms}}^2 t^2. \tag{3.33}$$

and the separation between the two particles is said to be *ballistic*, i.e. taking place at a constant rate.

Notice that the mean square displacement is expected to be isotropic because the forcing of the flow is itself isotropic. By measuring and comparing the displacements along the $x-$direction $\left\langle R_x^2(t) \right\rangle$ and along the $y-$direction $\left\langle R_y^2(t) \right\rangle$ (not shown here), these were observed to be identical $\forall t$. This equality does not fully ensures isotropy but strongly suggests it.

In order to evaluate (3.30) and (3.31) in the case of continuous particle emission in the inverse energy cascade, a numerical simulation is run at a spatial resolution $N_x^2 = 512^2$, with small-scale, delta-correlated forcing at $k_f \in [100, 105]$. A fixed number of 5×10^{-6} tracers are uniformly spread in the square domain of size $L^2 = (2\pi)^2$. No boundary condition to remove the particles are applied.

Figure 3.4 (left) displays $C(\tau)$ for various values of the Reynolds numbers Re_α. One can see that $C(\tau)$ is indeed a constant function with value unity at small times, falling to zero after a time comparable to the large eddies turn over time $T_\alpha = l_\alpha / u_{\mathrm{rms}}$.

Figure 3.4 (right) reports the mean square displacement $\left\langle R^2(t) \right\rangle$ as a function of time and for various Reynolds numbers, which was adjusted by varying the Ekman friction coefficient α. The ballistic regime is clearly seen at small times, and the agreement with the dotted-dashed line representing $u_{\mathrm{rms}}^2 t^2$ is excellent. The diffusive regime is also represented by the dashed line.

The lowest Reynolds used was actually chosen to be below the turbulent transition. It is nevertheless represented on the figures to stress that the necessary ingredient to obtain this succession of regimes - ballistic then diffusive - is only to have a finite Lagrangian correlation time, whatever the turbulent state of the flow.

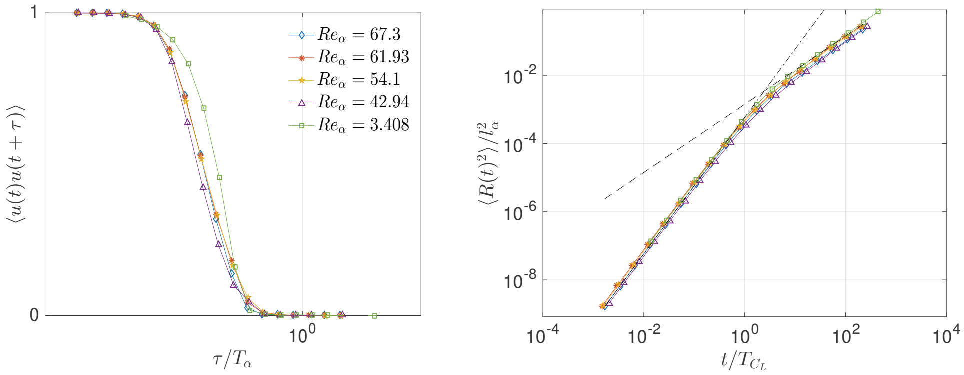


Figure 3.4: Left: Lagrangian velocity correlation functions $C(\tau)$ in log-lin plot. Curves are shown to superimpose when represented as a function of τ/T_α with $T_\alpha = l_\alpha/u_{\text{rms}}$. Right: mean square displacements as a function of time for the same Reynolds numbers. T_{C_L} is the Lagrangian correlation time. Dotted–dashed line represents the ballistic regime $\langle R^2\rangle = u_{\text{rms}}^2 t^2$ for the highest Reynolds which fits very well with the data. Dashed line represents the diffusive regime with $\langle R^2\rangle \propto t$.

These measures, together with the ones for the equilibrium density probability, confirm the validity of the ballistic and diffusive approximations for the tracer motion injected by a point source in the two-dimensional inverse cascade.

Equilibrium distribution

As stated above, when including absorbing boundary conditions, the total number of particles present in the domain reaches a statistically steady state, for which the average of N_p over a given time interval T is then $\langle N_p\rangle_T = \text{cst}$. One can wonder about the equilibrium distribution for the number of particles as a function of the distance from the source, $n(R)$.

As explained in the introduction of this chapter (see 3.1), point particle dispersion can be related to a scalar field $\theta(\boldsymbol{x})$. To obtain a theoretical prediction for this quantity, two regimes must be considered, either if we are close or far from the source.

For short travelled distances (or times from ejection), particles mainly see a constant velocity u_{rms}. They are thus shot radially from the source and distributed uniformly on an enclosing surface, i.e. a circle in two dimensions. For a perfect continuous emission, this ballistic regime creates a field of concentration

$$\theta(R) = \frac{\phi_S}{u_{\text{rms}} R} . \tag{3.34}$$

For large distances, the scalar mixing is expected to be dominated by diffusion as expressed by equation (3.22). Averaged over time, it yields the stationary solution $\langle\theta\rangle$:

$$0 = \kappa_T \nabla^2 \langle \theta \rangle + \phi_S \delta(\boldsymbol{x} - \boldsymbol{x}_S). \tag{3.35}$$

Integrating (3.35) on a disk of radius R centered on the source (denoted by Γ_R), then using the Stokes theorem and assuming isotropy, we get:

$$\int_{\Gamma_R} \kappa_T \boldsymbol{\nabla} \langle \theta \rangle \cdot \boldsymbol{e}_r = \int_{\Gamma_R} \kappa_T (\partial_R \langle \theta \rangle) R^{d-1} \mathrm{d}\Omega_d = -\phi_S, \tag{3.36}$$

where d is the dimension and $\mathrm{d}\Omega_d$ the solid angle element in d dimensions (for $d = 2$, it is an arc of a circle). Owing to isotropy:

$$\frac{\mathrm{d} \langle \theta \rangle}{\mathrm{d}R} = -\frac{\phi_S R^{1-d}}{\kappa_T 2\pi}. \tag{3.37}$$

For $d = 2$, the solution reads:

$$\langle \theta(R) \rangle = \langle \theta \rangle_0 - \frac{\phi_S \log R}{\kappa_T 2\pi}. \tag{3.38}$$

Absorption condition is $\theta(R_{\max}) = 0$. Then:

$$\langle \theta(R) \rangle = -\frac{\phi_S \log{(R/R_{\max})}}{2\pi \kappa_T}. \tag{3.39}$$

A time scale t^* at the crossover of these two regimes may be defined, i.e. beyond which the diffusive regime is recovered. t^* is associated to a scale R^* such that:

$$t^* = R^*/u_{\mathrm{rms}} = (R^*)^2/\kappa_T \Rightarrow R^* = \kappa_T/u_{\mathrm{rms}}. \tag{3.40}$$

To assess the agreement between this expectation in a turbulent inverse cascade, a numerical simulation is carried at resolution $N_x^2 = 512^2$ in a periodic square domain $[0\ 2\pi] \times [0\ 2\pi]$. Particles are seeded in a uniform region of size $l/l_f = 8 \times 10^{-2}$ surrounding the source situated at $\boldsymbol{x}_S = (\pi, \pi)$. 10^2 particles are injected per time step with $\Delta t = 10^{-3}$, hence a rate $\phi_S = 10^5$ particles per unit time.

The spatial absorbing boundary condition is at $R_{\max} = L/4 = \pi/2$. Statistical stationarity of the total number of particles in the domain is expected to be reached after a time $R_{\max}/u_{\mathrm{rms}}$. This was indeed verified numerically.

The simulation is run for another $\approx 900\, T_{R_{\max}}$, with $T_{R_{\max}} = R_{\max}/u_{\mathrm{rms}}$ the turn over time of the eddies of size $R_{\max}$. The profile $\langle \theta(R) \rangle$ is computed by averaging over time and the number of particles in annulus shells around the source.

$\langle \theta(R) \rangle$ is displayed in Figure 3.5. The left panel focuses on the density profile close to the source and is shown to be indeed decreasing as R^{-1}, in agreement with (3.34) and validating the ballistic approximation for short emission times. The right panel focuses on

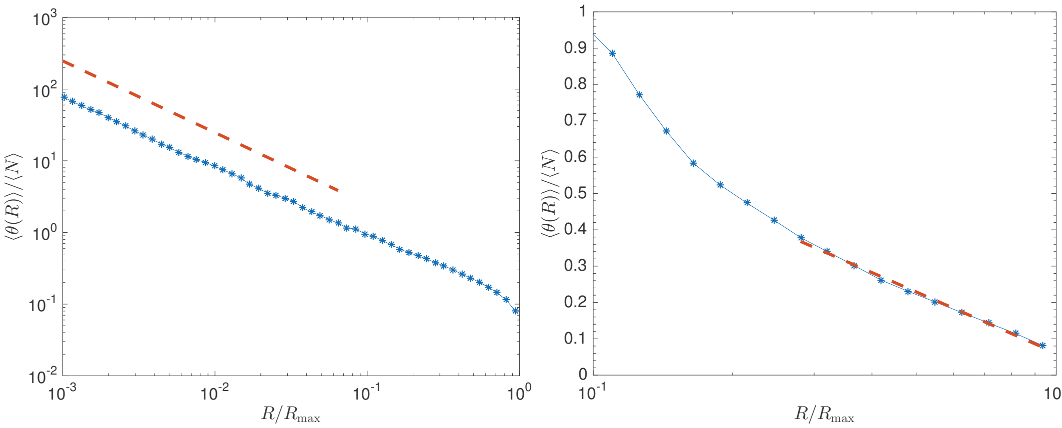


Figure 3.5: Average number density $\theta(R)$ as a function of the distance from the emission point, normalised by the average number of particles in the domain $\langle N\rangle = \frac{1}{R_{\max}}\int_0^{R_{\max}} N(R)\,\mathrm{d}R$. Left: focus on ballistic regime at short distances. Red dashed line corresponds to the solution 3.34. Right: zoom on large R the effect of the absorbing condition solution. Red dashed line is a fit over the large R using the form of the solution (3.39). The parameters of the fit may then be used to estimate the turbulent diffusivity κ_T.

the large distances, where the effect of the absorbing boundary condition creates a profile compatible with (3.39), thus ensuring the validity of the diffusion approximation (3.35).

We conclude this section by a remark, noting that statistical convergence requires long time averages. Indeed, instantaneous snapshots of the particle distribution are shown in Figure 3.6, each separated by a few integral time scales, revealing strong instantaneous inhomogeneities and anisotropies.

3.3.2 Two-point correlation

Figure 3.7 displays an instantaneous distribution of particles in the inverse energy cascade. It was obtained in a simulation at resolution $N_x^2 = 4096^2$ using the line injection mechanism (see section 3.2.2). Several qualitative comments can be made upon the observed features of the flow.

- Starting from the source (indicated by the red cross on the figure) we first note that the particles migrate away from the source following a nearly straight line. The nearly ballistic regime followed by particles as they separate from the source. They are transported by larger eddies and their trajectory is deformed by small vortices encountered.

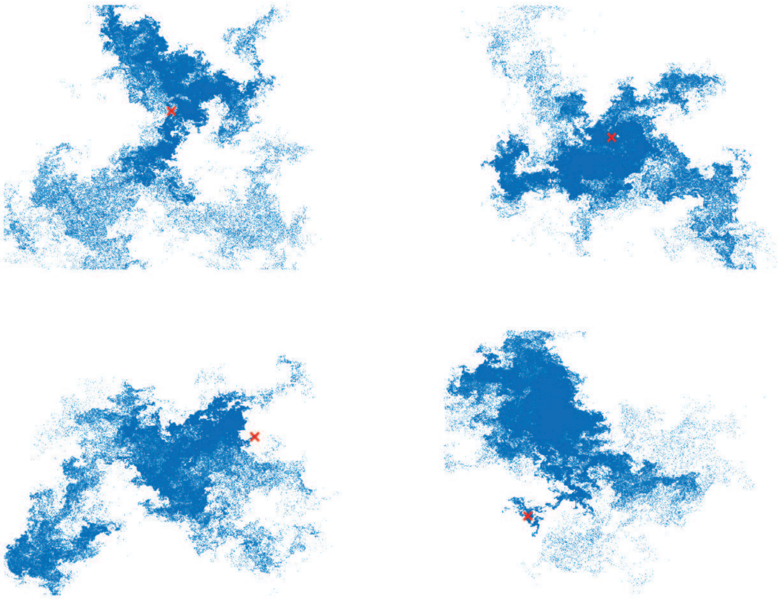


Figure 3.6: Instantaneous cloud distribution at times 2.5, 5, 7.5, 16 T_I. Strong inhomogeneities due to large eddies transport are observable. The width of the window is $L/6$. Red cross indicates the source.

- Further, strong and sudden deviations from this regime are revealed, where the line of particles is deformed into a sheet with increasing stretching rate along the transverse direction and compressing rate along the longitudinal direction. This is a manifestation place of *homoclinic tangle*, where an unstable and a stable manifold mix together. Under these circumstances, it is legitimate to expect fractality in the particle cloud (Manneville, 2010).
- Going further away from the source, complex patterns combine coherent lines folded by the dynamic, still observable at small scales, and a rather uniform background.
- The qualitative distribution obviously changes as we go even further away from the source. At the top of the figure, one can see that the cloud becomes more and more uniform. It is thus expected to recover a homogeneous mass distribution far from the source and at large scales.

Following this qualitative description, we wish to characterise the geometry of the distribution and how it varies with the distance R from the source. The complex mix of the different dynamical regimes and foldings of the line initially emitted gives the intuition of a multi-

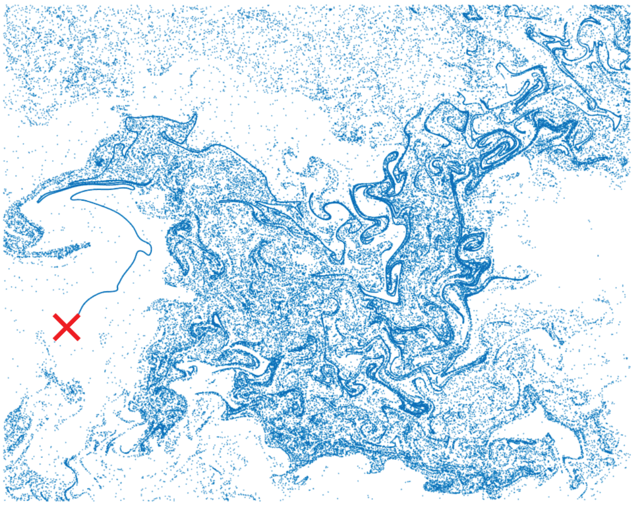


Figure 3.7: Zoom on the particle distribution around the source (red cross). The width of the window is $L/50 = 20\,l_f$.

fractal distribution (Beigie *et al.*, 1994). One way to measure the fractality of the ensemble is to determine the scaling of the quasi-Lagrangian mass $m_{\mathrm{QL}}(R,r) = \langle m_{\mathrm{QL}}(R,r,t)\rangle_t$, which relates to the average mass found, on average, around a particle in a ball of size r at a distance R. $m_{\mathrm{QL}}(R,r)$ actually corresponds to the two-point correlation

$$C_2(R,r) = \langle \theta(\boldsymbol{X}_1,t)\theta(\boldsymbol{X}_2,t)\rangle_t \tag{3.41}$$

integrated over a volume r, with $R = \left\| \frac{1}{2}(\boldsymbol{X}_1 + \boldsymbol{X}_2) - \boldsymbol{x}_S \right\|$ and $r = \|\boldsymbol{X}_1 - \boldsymbol{X}_2\|$, hence:

$$m_{\mathrm{QL}}(R,r) = \int_0^r C_2(R,r')\,\mathrm{d}r'. \tag{3.42}$$

This quantity can also be defined in terms of the two-point transition probability in the following way: let $n(\boldsymbol{x},t)$ be the number of particles at position $\boldsymbol{x}$ at time t. It is the contribution of all particles emitted in the past $t_0 < t$ and coming from any position $\boldsymbol{x}_0$.

Hence:

$$\begin{aligned}\langle n(\boldsymbol{x},t)\rangle &= \int\limits_{-\infty}^{t}\int\limits_{-\infty}^{\infty} \delta(\boldsymbol{x}-\boldsymbol{x}(t|\boldsymbol{x}_0,t_0))\phi_S\delta(\boldsymbol{x}_0-\boldsymbol{x}_S)\,\mathrm{d}t_0\mathrm{d}\boldsymbol{x}_0\\ &= \phi_S\int\limits_{-\infty} t\delta(\boldsymbol{x}-\boldsymbol{x}(t|\boldsymbol{x}_0,t_0))\,\mathrm{d}t_0\end{aligned}$$

The integral over the initial position vanishes because all particles originate from the same position $\boldsymbol{x}_S$ corresponding to the emitting source. The average product of this quantity at two different positions $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{y}$ is

$$\langle n(\boldsymbol{x},t)n(\boldsymbol{x}+\boldsymbol{y},t)\rangle = \phi_S^2\iint \mathrm{d}t_0\mathrm{d}t_0'\;P(\boldsymbol{x},t,\boldsymbol{x}+\boldsymbol{y},t|t_0,t_0') \tag{3.43}$$

The integrand in this last equation denotes the joint transition probability of finding a particle 1 at position $\boldsymbol{x}$ at time t emitted at time t_0 and a particle 2 at position $\boldsymbol{x}+\boldsymbol{y}$ at time t emitted at time t_0'. It can be decomposed as:

$$\langle n(\boldsymbol{x},t)n(\boldsymbol{x}+\boldsymbol{y},t)\rangle = \phi_S^2\int\limits_{-\infty}^{t}\mathrm{d}t_0P(\boldsymbol{x},t|t_0)\int\limits_{-\infty}^{t}\mathrm{d}t_0'\;P(\boldsymbol{x}+\boldsymbol{y},t|\boldsymbol{x},t,t_0,t_0') \tag{3.44}$$

Integrating over a ball of size r centered on $\boldsymbol{x}$ yields:

$$\int_{B_r}\mathrm{d}\boldsymbol{y}\;\langle n(\boldsymbol{x},t)n(\boldsymbol{x}+\boldsymbol{y},t)\rangle = \phi_S^2\int\mathrm{d}t_0P(\boldsymbol{x},t|t_0)\underbrace{\int_{B_r}\mathrm{d}\boldsymbol{y}\int\mathrm{d}t_0'\;P(\boldsymbol{x}+\boldsymbol{y},t|\boldsymbol{x},t,t_0,t_0')}_{m_{\mathrm{QL}}} \tag{3.45}$$

The under-braced term denotes the quasi-Lagrangian mass and is the object to be evaluated as a function of r.

We first chose to compute the correlation dimension $\mathcal{D}_2$, linked to pair probability. Its measure requires evaluating the correlation integral. In practice it is approximated by the discrete count of the number of pairs whose distance is smaller than r:

$$m_{\mathrm{QL}} = \sum_{i<j}\Theta(r-|\boldsymbol{X}_i-\boldsymbol{X}_j|) \tag{3.46}$$

with Θ the Heaviside function. Then $m_{\mathrm{QL}}\propto r^{\mathcal{D}_2}$ for $r\to 0$.

As discussed above, we expect the distribution to vary from a line ($\mathcal{D}_2=1$) close to the source to a uniform distribution ($\mathcal{D}_2=2$) far from the source, i.e:

$$\mathcal{D}_2 = \begin{cases} 1 & R\ll R^* \\ 2 & R\gg R^*\end{cases}$$

This kind of analysis requires a scaling range as large as possible. We thus performed high spatial resolution simulations at $N_x = 4096^2$. In order not to be contaminated by the recurrence problem (see section 3.1.2), we don't use an absorbing boundary condition at a given distance from the source but we rather fix a finite lifetime for the particles $t_{p_{\max}}$. The physical origin of this removal condition is that the concentration transported by fluid elements will gradually fade over time due to diffusion. After a given time, these fluid elements will contribute by only a negligible amount to the quasi-Lagrangian mass.

A numerical simulation is run for a duration of $\sim 10\, T_L$ in the statistically steady state for both fluid kinetic energy and particle number. With the criterion of removing particles after a fixed time $t_{p_{\max}}$, the number of particles in the domain is exactly constant with value $N = \phi_S t_{p_{\max}}$.

The region around the source is divided into annulus zones at distances R from the source. For each zone, the number of pairs that are separated from a distance below r are counted and averaged in time over $\sim$ 10 large-eddy turn over times. A very large number of scanned pairs is considered to reach sufficiently interpretable statistics. It reaches 5.12×10^{13} and the counting necessitates 5×10^3 equivalent CPU-hours for each value of $t_{p_{\max}}$.

For two particles $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, the probability that their inter-distance is less than r conditioned on the fact that their center of mass is at a distance R from the source reads:

$$p_2^<(r,R) = \mathcal{P}\left(\|\boldsymbol{X}_1 - \boldsymbol{X}_2\| < r \;\middle|\; \left\| \frac{(\boldsymbol{X}_1 + \boldsymbol{X}_2)}{2} - \boldsymbol{x}_S \right\| = R \right). \tag{3.47}$$

In practice, the condition on R is restricted to an annulus zone of width ΔR.

This pdf scales with the size r of the ball with an exponent:

$$D(r) = \frac{\mathrm{d}\log p_2^<(r)}{\mathrm{d}\log r}. \tag{3.48}$$

Formally, $D(r) \to \mathcal{D}_2$ for $r \to 0$. In practice, small scales are very under-sampled for finite sample sizes, and we look at the plot of $D(r)$ hoping to find a plateau over a sufficiently large scale range which would yield a reasonable estimation for $\mathcal{D}_2$. More details and alternatives on the methods to determine the correlation dimension may be found in Grassberger & Procaccia (2004); Takens (1985); Theiler (1988).

Figure 3.8 (left panel) shows the quantity $p_2^<(r)$ for various distances R from the source. For R small, we are able to recover a scaling $p_2^<(r) \propto r$ which yields $\mathcal{D}_2 = 1$, corresponding effectively to the emitted line. As the distance R increases, one can see that this linear distribution is still recovered at small scales, as was observed in figure 3.7. At larger scales, the slope increases, as was already our guess: the distribution becomes more and more uniform as one gets further from the source. The drop of the curves at very small scales r can be interpreted as a lack of statistics.

Figure 3.8 (right panel) depicts in more details the logarithmic slopes of $p_2^<(r)$ (equation (3.48)). Curves of $D(r)$ exhibit an increasing behaviour with the scale r, reaching a maximum at $1 - 10l_f$. Note that only data $r \ll \Delta R \sim 5l_f$ should be considered. Hence, in the of the falling curves for large r, corresponding to $r > \Delta R$, the value of R assigned to each line is not relevant any more. It is only shown for completeness and to illustrate that no clear plateau is observable. Rather, we get a smooth variation of the scaling exponent.

This variation actually comes from a contamination from a uniform background that is more and more dominant with increasing R. It has a marked effect on all scales r, as shown by an increase of $D(r)$ at large R even for small $r/l_f \ll 1$. The line emitted by the source is thus recovered at smaller and smaller r as R increases. Only close to the source ($R \gtrsim l_f$) is the line presence clearly marked.

To confront the hypotheses of increasing $D(r)$ because of continuing return from long-living particles, we did measure the same quantities, with the same geometrical framework, and varying the lifetime of the particles $t_{p_{\max}}$. If particles are allowed to live a shorter time, they contribute less to this uniformisation. One would thus expect the maximum of $D(r)$ to lower. Same quantities are shown in Figure 3.9. It is indeed seen that the maximum of $D(r)$ decreases with $t_{p_{\max}}$, and that the line $D(r) = 1$ is quickly recovered. Only data for an arbitrary $R = 52.5l_f$ is shown, but this effect was ensured to be present at all distances R. This validates our starting hypotheses, which makes impossible the estimation of a correlation dimension $\mathcal{D}_2$.

It thus appears that the difficulty to determine a unique, clear scaling for m_{QL} results from the combination of a large variety of physical mixing processes: ballistic transport, mixing by turbulent eddies, and uniformisation by long time diffusion. In the next section, another way to determine the scaling of the quasi-Lagrangian mass is proposed.

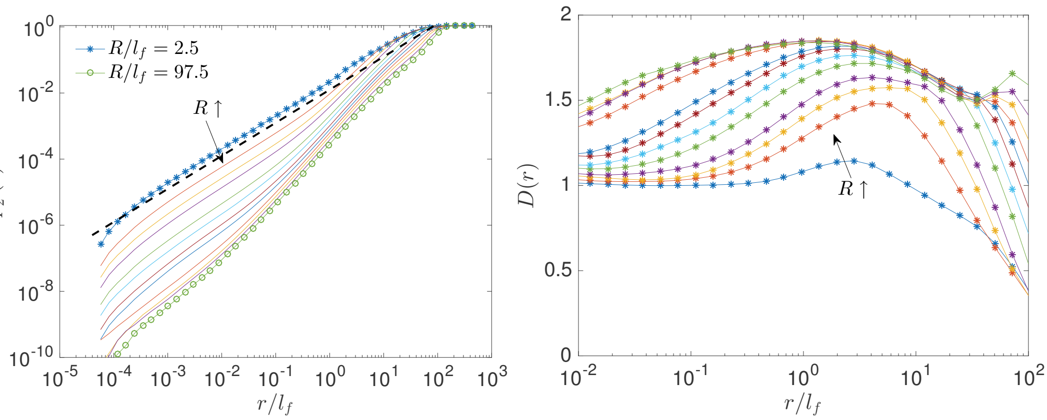


Figure 3.8: Left: $p_2^<(r)$ for various distances from the source R and $t_{p_{\max}} = 0.4$. Black dashed line is $\propto r$ and red dashed line $\propto r^{5/3}$. Right: details of the logarithmic slopes $D(r)$ for the curves on the left.

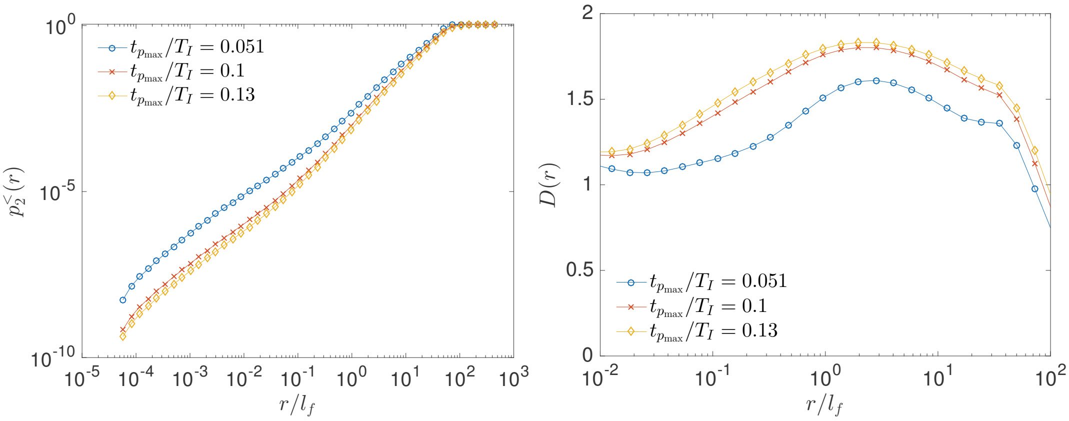


Figure 3.9: Same plots as in Figure 3.8 for a fixed value $R = 52.5\, l_f$ and varying $t_{p_{\max}}$. T_I is the integral time.

3.3.3 Phenomenological description

In this section, another approach is presented to describe the quasi-Lagrangian mass fluctuations in the emitted particle distribution.

Consider an emitted line Γ with a definite length L_C. Due to velocity increments fluctuations, it will undergo longitudinal stretching and compressing, in addition to bending, and we may define a line density profile $\rho(s,t)$. The total mass of this line is $m = \int_\Gamma \rho(s,t)\, \mathrm{d}s(t)$ and grows linearly in time $m(t) = \phi_S\, t$, if no removal term is included.

We wish to exploit these continuous stretchings and foldings in space to quantify the total mass present in a region of size r at a distance R from the source. Figure 3.10 shows a scheme introducing the notations used hereafter. The idea is the following: a first reference particle P_0 at a distance R_0 from the source and an age A_0 is picked. Among older ones, a particle P_H is searched corresponding to the last one to be inside a circle of radius r around P_0 before the line escapes further away from P_0. In practice, this distance is set to $2r$. This limits the very short returns inside the ball, adding a small error r at the determination of the distance R_0. The age of P_H is noted A_H. The line joining particles P_0 and P_H is represented in light green on the scheme.

The average age difference between particles P_0 and P_H is written as:

$$\left\langle \Delta A_{H0}^{(i)} \right\rangle = \left\langle A_H^{(i)} - A_0^{(i)} \right\rangle \tag{3.49}$$

and is a function of R_0, A_0 and r. The superscript i denotes the index of the line. The first line just described above has the index $i = 0$. When this line comes back at a distance below $\leq r$ from particle P_0, the particle P_R is marked. It then serves as the new reference particle and the algorithm is repeated. The new line then has the index $i = 1$. In the following, this superscript is not written except where needed.

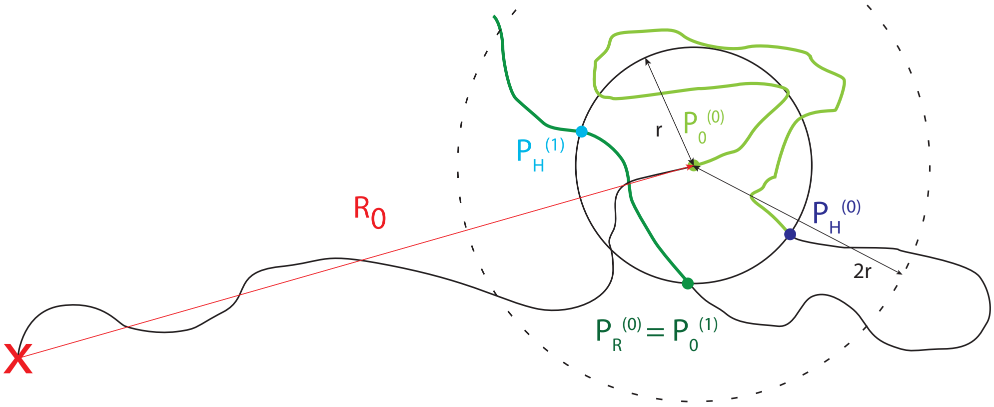


Figure 3.10: Scheme of the setup based on particle ages to determine quasi-Lagrangian mass scaling. The red cross represents the source from which particles are emitted. The contributions of the mass around a reference particle P_0 inside the circle of radius r come from the green portions of the line.

The quantity (3.49) is a measure of the (average) mass between particles P_0 and P_H. Indeed, it corresponds to what is emitted during a time $\langle \Delta A_{H0} \rangle$, hence $m = \phi_S \, \langle \Delta A_{H0} \rangle$. The quasi-Lagrangian mass may then be viewed as the total contribution from all these lines inside the ball of radius r:

$$m_{QL} = \phi_S \sum_{i}^{\infty} \left\langle A_H^{(i)} - A_0^{(i)} \right\rangle . \tag{3.50}$$

Theoretical predictions

Multiple regimes can already be guessed using what is known about relative dispersion in turbulent flow (see section 2.3). Let $r(t)$ be the distance between particle P_0 and P_H at time t and r_E the distance between these same particles at the moment t_E, when the particle P_0 is emitted. r_E is thus the distance that the elder particle P_H will have travelled from the source before particle P_0 is emitted. This separation is of the order of $r_E = u_{\mathrm{rms}} \Delta A_{H0}$.

Different cases described here below can be distinguished. Notice that the characteristic length scale used to compare r_E is the forcing length scale l_f. In our two dimensional simulations focusing on the inverse energy cascade, i.e. with a very small scale forcing to maximise the inertial range, this is tantamount to compare r_E with the dissipative scale l_D. Below l_f, the flow is actually differentiable.

- Case 1: $r_E \ll l_f$

  - Case 1a: $r(t) \ll l_f$
    The distance between the two particles always remains in the sub-forcing scales, hence in the dissipative regime of the flow, so that $r(t) = r_E e^{\xi t}$, where $\xi(t, r_E)$ is the positive finite-time Lyapunov exponent, thus $r(t) \propto \Delta A_{H0}$.
  - Case 1b: $r(t) \gg l_f$
    The separation can be divided in two steps. Firstly, initial separation has started in the dissipative regime and lasts until the separation equals the forcing scale: $r(t) = l_f = r_E \exp(\lambda t^*)$, hence during a time $t^* = \frac{1}{\lambda} \log \frac{l_f}{r_E}$. Secondly, once their distance is above the forcing scale, separation will be ballistic during a turn-over time associated with the scale l_f,τ_{r_E}, and will after be dominated by the Richardson explosive regime : $r(t) = \epsilon^{1/2}(A_0 - t^*)^{3/2}$. From this relation, we have that $r_E \propto l_f e^{r(t)^{2/3} - A_0}$. Hence, for $A_0 \gg r^{2/3}\epsilon{-}1/3$, the quantity ΔA_{H0} is independent of r.

- Case 2: $r_E \gg l_f$

Case 2a: $A_0 < \epsilon^{-1/3} r_E^{2/3} = \tau_{r_E}$
Particles P_0 and P_H are initially distant from $r_E > l_f$. They separate during a time A_0 which is inferior to the eddy turn-over time associated to their initial separation τ_{r_E}. This corresponds to the ballistic regime prior to Richardson explosive separation, where particles see a constant velocity difference $\delta_{r_E} u$, so that $r(t) = r_E + A_0 \delta_{r_E} u$. For $A_0 \ll \Delta A_{H0} u_{\mathrm{rms}} / \delta_{r_E} u$, it is thus also expected to recover $\langle \Delta A_{H0} \rangle \propto r$ as in case 1a.

Case 2b: $A_0 > \epsilon^{-1/3} r_E^{2/3} = \tau_{r_E}$
If particles have separated during a time A_0 longer than τ_{r_E}, the inter-distance is then in the explosive Richardson regime: $r(t) \propto A_0^{3/2}$ and is independent of the initial separation r_E.

Figure 3.11 (left) summarises the relations listed above.

Numerical measurements

To measure the quantity ΔA_{H0}, a numerical simulation at resolution $N_x = 4096^2$ is performed in which particles are emitted from the source situated at position $\boldsymbol{x}_S = [\pi, \pi]$, with a maximum lifetime for the particles $t_{p_{\max}} = \frac{L}{2u_{\mathrm{rms}}}$, where $L = 2\pi$ is the size of the square domain. Table 3.1 gives the parameters used for the simulation as well as global flow quantities.

Because we simulate discrete particles, we will of course not find P_H at an exact distance r from P_0. Limiting ourselves to the closest particle from the theoretical P_H will then yield an error on A_H and on ΔA_{H0}. The maximum precision we can obtain for A_H is equal to the time step Δt because one particle is released per time step. One way to get more precision

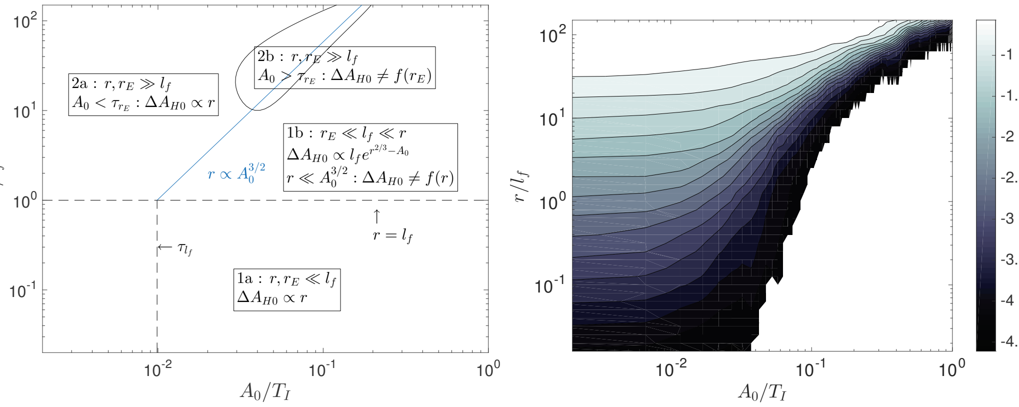


Figure 3.11: Left: scheme of expected regimes for $\langle \Delta A_{H0} \rangle$. Notation $\neq f(\cdot)$ means: "is not a function of". Right: contour plot of $\log_{10} \langle \Delta A_{H0} \rangle$. In the simulation, the condition $r_E \gg l_f \Leftrightarrow \Delta A_{H0} \gg l_f/u_{\mathrm{rms}} = 4 \times 10^{-3}$.

ν	Δt	ϕ_S	L	T_I	u_{rms}	ε	k_f	l_f	t_{l_f}
4×10^{-4}	10^{-5}	10^5	2π	1.96	1.6	41.6	1000	$2\pi \times 10^{-3}$	0.01

Table 3.1: Some parameters of the simulation. ν is the fluid molecular viscosity. Δt is the time step. ϕ_S is the injection rate in units of particles injected per time step. $L = 2\pi$ is the size of the square domain. $T_I = L/u_{\mathrm{rms}}$ is the time associated to the integral scales. $u_{\mathrm{rms}} = \sqrt{2E}$ is the mean square velocity. $\varepsilon = -2\nu Z$. k_f is the forcing wave-number. $l_f = 10^{-3} L$ is the forcing scale. $t_{l_f} = \varepsilon^{-1/3} l_f^{2/3}$ is the time associated to the forcing scale.

could be to increase the number of emitted particles per time step and to interpolate linearly their age between the emission time t_E and $t_E - \Delta t$. This would however increase the numerical cost, and another method is implemented.

Let $D(P_A, P_B)$ be the distance between particles P_A and P_B. Then particle P_H corresponds to a fictive point such that $D(P_0, P_H) = r$ whose age is linearly interpolated between particles, say P_{H_1} and P_{H_2}, which are directly surrounding particle P_H, i.e for which $D(P_0, P_{H_1}) < r$ and $D(P_0, P_{H_2}) > r$. This is tantamount to suppose that particles in the continuous framework are aligned between P_{H_1} and P_{H_2}, or equivalently that the line is not deformed on a time scale Δt. This is a valid hypotheses if $D(P_{H_1}, P_{H_2})$ is small compared to the correlation length of the flow, and if Δt is small compared to the temporal correlation of velocity increment associated with the scale $D(P_{H_1}, P_{H_2})$.

The determination of A_H then reads:

$$A_H = A_{H_1} + (A_{H_2} - A_{H_1}) \frac{\sqrt{\|\boldsymbol{r}_H - \boldsymbol{r}_{H_2}\|^2 + \|\boldsymbol{r}_H - \boldsymbol{r}_{H_2}\|^2 - 2\frac{\boldsymbol{r}_H \cdot \boldsymbol{r}_{H_1}}{\|r_H\| \|r_{H_1}\|}}}{\|\boldsymbol{r}_{H_1} - \boldsymbol{r}_{H_2}\|} \tag{3.51}$$

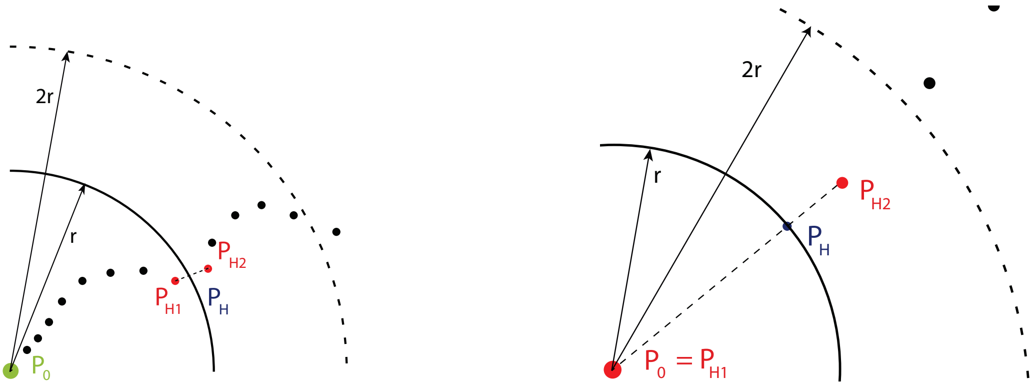


Figure 3.12: Left: the position of the true particle P_H has to be interpolated between two adjacent discrete particles. Right: For $A_0 \gg T_I$ and / or $r \ll u_{\rm rms}\Delta t$, the low spatial resolution of particles along the line yields a very bad estimation of its length.

For $A_0 \gg T_I$ and / or $r \ll u_{\rm rms}\Delta t$, particles separate quickly and the distance r and $2r$ are rapidly attained. In that case, particle P_0 corresponds to P_{H_1} and particles P_0, P_H and P_{H_1} are aligned (see Figure 3.12). In this situation, $\langle \Delta A_{H0} \rangle$ trivially scales as r. This situation corresponds to a serious breaking of the continuous approximation at the scale r, and these events are removed from the statistics. As a consequence, values of $\langle \Delta A_{H0} \rangle$ will be lower-bounded by Δt.

The simulation is run for 2 T_I after the number of particles has reached its equilibrium. The quantity ΔA_{H0} is determined for all A_0 and various r logarithmically spaced between $10^{-4} = 20\pi l_f$ and $1 = L/2\pi$. This is performed for 200 different instantaneous distributions of tracers over which it is averaged.

We now wish to verify the theoretical predictions given above. Figure 3.13 (left panel) shows the age difference $\langle A_{H_0} \rangle / T_I$ as a function of the distance r/l_f for various A_0. Relation 1a, $\langle \Delta A_{H0} \rangle \propto r$, is indeed recovered in a wide range of scales for $A_0 \ll T_I$ and $r \ll l_f$. As A_0 grows, the asymptotic regime for $A_0 \gg r^{2/3}$, case 1b, is recovered, and $\langle \Delta A_{H0} \rangle$ is independent of r. The breaking of the lines correspond to the break-up of the continuity of the line at the scale r (see Figure 3.12 (right)).

Figure 3.13 (right panel) displays the same quantity as a function of $(r/l_f)/A_0^{3/2}$. In the explosive Richardson separation, $r(t) \propto A_0^{3/2}$ is a quantity independent of the initial separation r_E, or equivalently that $\langle \Delta A_{H0} \rangle \neq f(A_0)$. This is verified in our measures where such a point is indeed observable in Figure 3.13 at $((r/l_f)/A_0^{3/2}, \langle \Delta A_{H0} \rangle) \approx (2 \times 10^2, 2 \times 10^{-3})$.

A contour plot of the quantity $\log \langle \Delta A_{H0} \rangle$ represents these various regimes along with the theoretical predictions in Figure 3.11 (right).

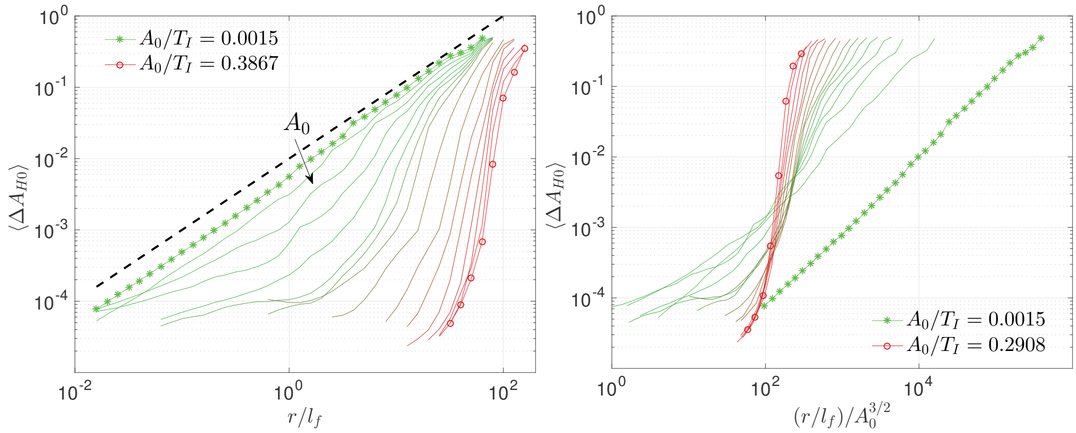


Figure 3.13: Age difference between particles separated from a distance r for various ages A_0 of the reference particle. The range of A_0 spans the whole values available until T_I, monotonically increasing in the direction indicated by the arrow. The black dashed line has a slope 1 and represents $\Delta A_{H0} \propto r$.

3.4 Brief conclusion

We have analysed the dispersion of tracer particles continuously emitted from a point source. This issue targets various natural phenomena such as the release of polluting species at ocean surfaces. The turbulent mixing following the release results from a complex combination. Numerical simulations of two-dimensional turbulence have been carried. One point quantities, such as mean square displacements and average radial concentration profiles were shown to obey simple ballistic dynamic at short times from emission and diffusion after one Lagrangian velocity correlation time scale. Then, we observed that the combination of various mixing processes, involving particles of very different ages, doesn't allow for the description of mass fluctuations with fractal dimensions. Another phenomenological approach was then proposed to account for the mixing of temporal and spatial correlations. It makes use of the knowledge of relative pair separation as a function of particles emission time and initial separation.

Part II

Inertial particle-laden flows

CHAPTER 4

A lattice method for the numerical modelling of inertial particles

Contents

4.1 Inertial particles dynamics

4.1.1 Individual particles

Massive particles, i.e. whose density differs from that of the fluid, experience a large variety of forces exerted by the carrier flow. Considering large particles requires to integrate the total constrain exerted by the fluid from the non-linear Navier-Stokes equations on the surface of the particle. This requires integrating analytically the full velocity field, which is in general not possible. However, an explicit expression for the total force can be derived in the case of small particles, i.e. smaller than the smallest characteristic length scale of the flow, and in the case of small velocity difference with surrounding flow. The flow around the particle is then in a laminar state, and the Reynolds number associated to the particle, $Re_p = d_p \langle V_p - u \rangle / \nu$, with d_p is the particle diameter and ν the fluid viscosity, is low. The non-linear term in the Navier-Stokes equation may then be neglected and the Stokes equation integrated around the particle (see Maxey & Riley (1983) for details).

This leads to a closed equation that we rewrite here in order to illustrate the complexity of the forces acting on each individual particle in a velocity field $\boldsymbol{u}(\boldsymbol{x})$:

$$\begin{aligned} m_p \frac{\mathrm{d}\boldsymbol{V}_p}{\mathrm{d}t} = {} & (m_p - m_f) g + m_f \frac{D\boldsymbol{u}}{Dt}(\boldsymbol{X}_p) - \frac{1}{2} m_f \left(\frac{\mathrm{d}\boldsymbol{V}_p}{\mathrm{d}t} - \frac{D\boldsymbol{u}}{Dt}(\boldsymbol{X}_p) \right) \\ & - 6\pi d_p^2 \rho_f \nu \int_0^t \frac{\mathrm{d}\tau}{\sqrt{\pi\nu(t-\tau)}} \frac{\mathrm{d}}{\mathrm{d}\tau} \left(\boldsymbol{v}_p - \boldsymbol{u}(\boldsymbol{X}_p(\tau), \tau) \right) \\ & - 6\pi d_p \rho_f \nu \left(\boldsymbol{V}_p - \boldsymbol{u}(\boldsymbol{X}_p) \right). \end{aligned} \tag{4.1}$$

m_p denotes the mass of each individual particle, $\boldsymbol{V}_p$ its velocity and $\boldsymbol{X}_p$ its position. ν is the kinematic viscosity of the carrier flow and m_f displaced mass of fluid by the particle.

The first term on the right-hand side is the buoyancy force. The second is the acceleration of the unperturbed flow at the particle position. The third one is the inertial correction, which arises because of the displaced flow by the particle, accounting for an additional transported mass. The fourth term is the *Basset-Boussinesq* history force, which obviously add a complicated effect linked to the past of the trajectory. It is due to the particle wake which acts to diffuse the flow vorticity apart from the particle trajectory. It is generally neglected when considering very small particles, because in that case the wake is dissipated on a sufficiently short length scale. The last term is the Stokes viscous drag. In this study, gravity is neglected, which is the case when fluid accelerations are stronger than g and particles not too massive.

With the simplifications cited above, one gets:

$$\frac{\mathrm{d}\boldsymbol{V}_p}{\mathrm{d}t} = \beta \frac{D\boldsymbol{u}}{Dt}(\boldsymbol{X}_p(t), t) - \frac{1}{\tau_p} \left[\boldsymbol{V}_p - \boldsymbol{u}(\boldsymbol{X}_p(t), t) \right]. \tag{4.2}$$

The first term on the right is the added-mass factor, with $\beta = 3\rho_f/(\rho_f + 2\rho_p)$. For heavy particles ($\rho_p \gg \rho_f$), we get $\beta \ll 1$ and this term is also neglected. The parameter $\tau_p = d_p^2/(3\beta\nu)$ denotes the response time of the particles. For $\rho_p \gg \rho_f$, $\tau_p = 2\rho_p d_p^2/(9\rho_f \nu)$. Finally, the dynamic of one particle, $\boldsymbol{X}_p(t)$, obeys the following equations:

$$\dot{\boldsymbol{X}}_p(t) = \boldsymbol{V_p}(t), \tag{4.3}$$

$$\dot{\boldsymbol{V}}_p(t) = -\frac{1}{\tau_p} \left[\boldsymbol{V_p}(\boldsymbol{X_p}, t) - \boldsymbol{u}(\boldsymbol{X_p}, t) \right]. \tag{4.4}$$

The dots denote temporal derivatives.

The non-dimensional number quantifying the relative inertia, the Stokes number, is built by the ratio of this response time and a relevant characteristic timescale of the flow:

$$St = \frac{\tau_p}{\tau_f} \tag{4.5}$$

Fundamental differences characterise the asymptotics of very low and large inertia. One noticeable fact about inertial particles in fluids, at least for moderate inertia, is that they exhibit clustering, i.e. they concentrate in regions of given carrier flow topology. The knowledge, quantification and prediction of such clustering is of central importance when considering situations where interactions between particles play a key role, such as in coalescence, or advection-reaction (Bodenschatz *et al.*, 2010; Krstulovic *et al.*, 2013).

The dynamic described by (4.3) and 4.4 is dissipative, with an associated phase-space contraction rate of $-d/\tau_p$. It has been shown in Bec (2003) that the long time behaviour of those particles converge toward a multifractal set.

St ≪ 1

In the limit when $St \to 0$, particles behave like tracers, and the difference $V - u$ in (4.4) cancels. One may then track individual particles in a Lagrangian framework, or consider solely a transport equation for their density $\rho_p(\boldsymbol{x})$ by the velocity $\boldsymbol{v}_p(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})$. The fractal dimension of the ensemble is the space dimension $\mathcal{D} = d$: the ensemble is homogeneous in space.

For very small response time (i.e. fast relaxation), the same technique may be used, but with a small correction applied to $\boldsymbol{v}_p$ of the order of τ_p (Maxey, 1987):

$$\boldsymbol{v}_p = \boldsymbol{u} - \tau_p \frac{D\boldsymbol{u}}{Dt}. \tag{4.6}$$

which appeared to yield good results compared to Lagrangian simulation for up to $St \approx 0.2$ (see, for instance, Shotorban & Balachandar (2006)).

Increasing St, trajectories of particles and fluid elements begin to separate and an evolution equation for the particle density must be provided. This density $\rho_p(\boldsymbol{x})$ is transported by a *compressible* flow (Balkovsky *et al.*, 2001) and its evolution, along with the field $\boldsymbol{v}_p(\boldsymbol{x})$, obeys the following dynamical system:

$$\partial_t \rho_p + \nabla \cdot (\boldsymbol{v}_p \rho_p) = 0, \tag{4.7}$$

$$\boldsymbol{v}_p = \boldsymbol{u} + \tau_p \left(\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u}\right). \tag{4.8}$$

This formulation is valid for Stokes up to $St \sim \mathcal{O}(1)$ (Balachandar & Eaton, 2010), and is to be preferred when particles are not in close equilibrium with the carrier phase, for example when they are injected perpendicularly to the flow. The divergence of the velocity field is given by (Maxey, 1987) :

$$\boldsymbol{\nabla}\cdot\boldsymbol{v}_p = \frac{-1}{4A}\left\{\underbrace{\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)^2}_{\text{strain rate}} - \underbrace{\left(\frac{\partial u_i}{\partial x_j}-\frac{\partial u_j}{\partial x_i}\right)^2}_{\text{vorticity}}\right\}. \tag{4.9}$$

with A a dimensionless parameter scaling as St^{-1}. The relation (4.9) indicates that the particles converge (the divergence is negative) where the strain dominates the vorticity, resulting in a preferential sampling of the high-strain regions.

This effect increases with the Stokes number. Indeed, when looking at an instantaneous spatial repartition of the particles, those leave near-empty regions whose size increases with St (Goto & Vassilicos, 2006). However, this increase in size is not only due to the centrifugal effect. In two dimensions, it has been shown that, for all $St < T/\tau_\eta$, where T is the characteristic timescale associated with the sweeping by the large eddies and $\tau_e ta$ the one associated to the dissipation scale, small inertial particles move away from non-zero acceleration zones. The clustering is better described by the correlation between the high-valued particle number density regions and the zero acceleration stagnation points (Chen *et al.*, 2006*a*).

Equations (4.8) and (4.7) have been successfully applied to study the dynamical properties of weakly inertial particles. For example in Boffetta *et al.* (2007), they are used to determine fractal dimension of the phase space attractor and yielded good agreement with Lagrangian simulations.

$\mathbf{St \gg 1}$

When the relaxation time gets larger, the existence and the uniqueness of the velocity field $\boldsymbol{v}_p(\boldsymbol{x})$ is not guaranteed any more. Indeed, inertia causes particles to detach from fluid trajectories with the possibility to cross each other with different velocities at the same position $\boldsymbol{x}$, forming *caustics*. Actually, this phenomenon already occurs for $St \sim \mathcal{O}(1)$. Figure 4.1 illustrates this phenomenon schematically in one dimension for a continuous line of particle. Obviously, this situation is predominant in the case where the fluid exhibits large and persistent gradients $\nabla\boldsymbol{u}$. The rate of formation of those caustics depends also on St. For example, in the regime of $St \ll 1$, it has actually been shown that the rate at which they form is $\propto \exp(-1/(6St))$ for $\tau_p \ll T \ll \tau_p \exp(1/6St))$ (Derevyanko *et al.*, 2007), thus showing an rapid decrease in the limit of vanishing inertia.

This particularity of inertial particles forces one the describe the particle dynamics in a higher-dimensional space.

In the limit $St \to \infty$, the motion is totally ballistic, with $\boldsymbol{V}_p$ independent of $\boldsymbol{u}$. The fractal dimension is $\mathcal{D} = 2d$, and particles occupy the full position-velocity phase-space.

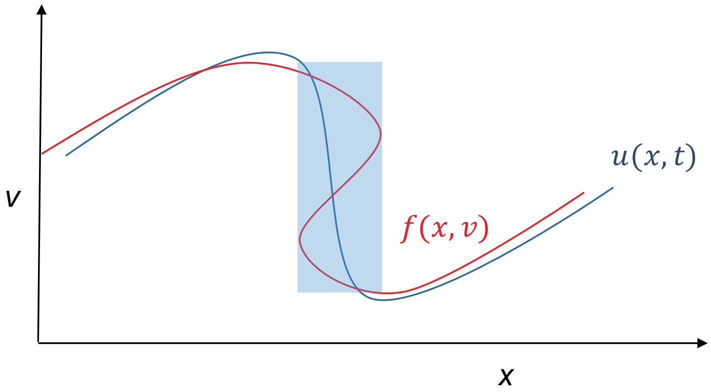


Figure 4.1: Illustration in one dimension of caustic formation: when inertia is large, the particle density f may *overshoot* the fluid velocity profile u and form multivalued velocity regions (blue-shaded on the figure).

In between the asymptotics $St \to 0$ and $St \to \infty$, a maximum of clustering can be found, characterised by a minimum fractal dimension. This is depicted by Figure 4.2 (right), where the dimension between the Lyapunov and the space dimension is represented as a function of St for $d = 2$ and $d = 3$. Figure 4.2 (left) illustrates the particles distribution near the maximum of clustering ($St \sim 0.2$).

An analysis of the domain of validity for each approximation in terms of St may be found in (Balachandar & Eaton, 2010).

4.2 The modelling of dispersed multiphase flows

Many processing technologies require good analysis of their capabilities and performance. Examples include cavitating pumps, papermaking, fluidized beds, etc. Very few processes involving material transport don't benefit from a better understanding of a multiphase dynamics. The definition of multiphase flows encompasses diverse cases, such as an arbitrary number of fluid mixtures, or solid suspensions. In addition, suspensions may involve particles of the same kind but of unique size (we then talk about *monodisperse* suspensions) or of various sizes (*polydisperse*). To add up in difficulty, coupling between different phases may also be considered. This increases the degree of complexity depending on the number of phases and the nature of their interactions. Furthermore, the presence of a dispersed phase can make appear different scale-resolving issues that would not be present when considering the fluid phase alone. Think for example of bubbly flows, where bubbles may leave along their trajectory a turbulent wake, while the large-scale flow is actually laminar (Mudde *et al.*, 2008).

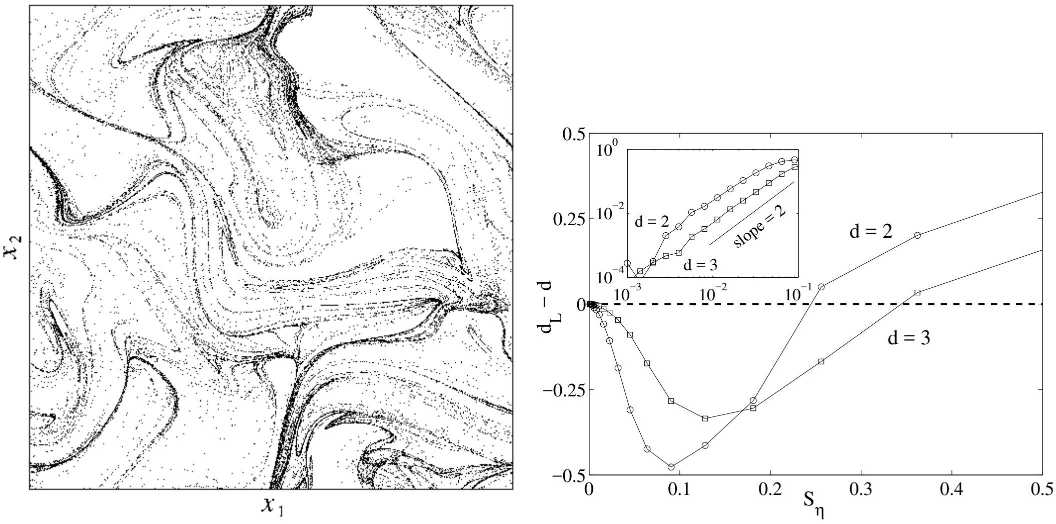


Figure 4.2: Left: Illustration of inertial particles distribution in position space. $St = 10^{-2}$. From Bec (2005). Right: difference between fractal dimension of the particle set and the space dimension d for increasing St (from Bec (2003)).

All of those difficulties add up, taking into account that most natural and industrial flows are in a turbulent state. And already, direct numerical simulations of a single turbulent phase need to resolve a tremendously large scale separation to reach high Reynolds numbers (up to 10^6 in some fluidized-beds). For all those reasons, the direct numerical simulation (DNS) of realistic multiphase flows is non attainable with any nowadays computational performance. Nevertheless, they constitute a required tool to better understand physical mechanisms and thus improving and validating the approximations made. As an example, models for polydisperse suspensions benefit from the knowledge of the interaction between particles of different sizes (Tenneti *et al.*, 2010; Yin & Sundaresan, 2009).

Such difficulties lead scientists and engineers to develop more and more sophisticated models and this section is dedicated to introduce briefly some of them.

4.2.1 From microscopic description to macroscopic quantities

Basically, one can think about two classes of methods to model multiphase flows. One is fundamentally relying on the detailed physical interactions between solid and fluid particles, or between particles and solid interfaces. The other is more phenomenological, based on conservation laws and constitutes a hydrodynamic approximation in which all relevant quantities are treated as continuous space fields. The term Lagrangian is used for models considering the evolution and tracking of discrete elements, while the term Eulerian is used for continuous fields approaches.

The transition between these two descriptions can be achieved for example through ensemble averaging: microscopic dynamical equations are volume or ensemble averaged to yield dynamical equations for the *low order macroscopic quantities*, such as mass or momentum (Drew & Passman, 1999).

In between microscopic and macroscopic models come the *kinetic equations* whose solution yields a function f defined over a phase space of relevant mesoscopic variables (volume, velocity, elasticity, etc.). Approximations are made from the microscopic description to model physical interactions. Detailed physical studies and direct numerical simulations are used to determine how the mesoscale variable associated to a given particle will be affected by the external forces, fluid, other particles (collisions), etc.

The moments of the kinetic equation allow then for recovering the macroscopic conservation equations. A famous example of kinetic equation is the one for gas dynamics or Boltzmann equation (Cercignani, 1988).

In this chapter, the collisions between solid particles are neglected and the mesoscopic variables are limited to position and velocity.

Lagrangian simulations

A solution to integrate the particle dynamics is to resolve explicitly equations (4.3) and (4.4) by considering point particles. This method presents two big advantages: it is not restricted in terms of St, and because each point is treated individually, polydispersity may be included easily. However, for a too-large number of particles in high-Reynolds-number turbulent flows, the number of degree of freedom is huge and models must be introduced. Another limitation of this approach is the difficulty to design numerical methods for the back-reaction from the solid phase on the continuous phase. Indeed, one needs a large number of particles and several approximations to handle correctly the distribution of the particles reaction on the numerical grid-points of the continuous phase. This issue is discussed in chapter 5. It was actually one of the main motivation for the development of the Eulerian description of inertial particles presented in this chapter, as it yields a more natural treatment of particle retro-action on the fluid.

When the number of particles becomes too large, so that their evolution is not computationally tractable, other stochastic approaches have to be considered. One of them is to build parcels which can be thought of as encompassing a large number of the discrete particles. These parcels have properties driven by a set of stochastic differential equations and their evolution represent Monte-Carlo simulations of the underlying pdf.

Their properties form a state vector $\boldsymbol{L}$ obeying a Langevin equation. For example, in the simplest case for which $\boldsymbol{L} = (\boldsymbol{X}_p, \boldsymbol{V}_p, \boldsymbol{u}_s(\boldsymbol{X}_p))$, where $\boldsymbol{u}_s$ is a velocity seen by the parcel, the equation takes the following form (Minier & Peirano, 2001):

$$
\begin{aligned}
\mathrm{d}\boldsymbol{X}_p &= \boldsymbol{V}_p \mathrm{d}t, && (4.10)\\
\mathrm{d}\boldsymbol{V}_p &= \boldsymbol{g}(t, \boldsymbol{L})\mathrm{d}t, && (4.11)\\
\mathrm{d}u_i &= h_i(t, \boldsymbol{L})\mathrm{d}t + \sum_j B_{ij}(t, \boldsymbol{L})\mathrm{d}W_j, && (4.12)
\end{aligned}
$$

where $\boldsymbol{g}$ is in our case the Stokes drag, $\boldsymbol{h}$ models the interaction between u_s and the mean fluid velocity $\langle u \rangle$. B is the Wiener process amplitude encompassing the model for the fluctuating part of the turbulent fluid velocity, like a $k - \epsilon$ model. The importance of the consistency between the turbulence models for the fluid and solid phases has been stressed in Chibbaro & Minier (2011).

These methods are largely applied in the engineering community. For example, Chibbaro & Minier (2008) have used this approach coupled to RANS equations for the fluid phase to study particle wall deposition and obtained a good agreement with experimental results.

Phase space description

The kinetic formulation for a particle population allows to encompass the physical properties of the particles (position, velocity...) in terms of an Eulerian field in the phase space:

$$
f(\boldsymbol{x}, \boldsymbol{v}, t) = \sum_p \boldsymbol{\delta}(\boldsymbol{X}_p(t) - \boldsymbol{x})\boldsymbol{\delta}(\boldsymbol{V}_p(t) - \boldsymbol{v}), \qquad (4.13)
$$

where δ is the delta function. With such a definition, the units of f are $L^{-2d}T^d$.

The evolution equation for f may be obtained the following way. f being distributional, we make use of a test function. Let $\varphi \in C^\infty$ be such a test function, infinitely smooth and with compact support. Then:

$$\begin{aligned}
&\partial_t \int f(\boldsymbol{x},\boldsymbol{v},t)\varphi(\boldsymbol{x},\boldsymbol{v})\ \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{v}\\
&= \partial_t \left[\sum_p \varphi(\boldsymbol{X}_p(t),\boldsymbol{V}_p(t))\right]\\
&= \sum_p \left[\dot{\boldsymbol{X}}_p(t)\cdot\boldsymbol{\nabla}_x\varphi + \dot{\boldsymbol{V}}_p(t)\cdot\boldsymbol{\nabla}_v\varphi\right]\\
&= \sum_p \left[\boldsymbol{V}_p(t)\cdot\boldsymbol{\nabla}_x\varphi + \boldsymbol{A}_p(t)\cdot\boldsymbol{\nabla}_v\varphi\right]\\
&= \sum_p \int \boldsymbol{v}\cdot\boldsymbol{\nabla}_x\varphi(\boldsymbol{x},\boldsymbol{v})\delta(\boldsymbol{x}-\boldsymbol{X}_p)\delta(\boldsymbol{v}-\boldsymbol{V}_p)\ \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{v}\\
&+ \sum_p \int \boldsymbol{a}\cdot\boldsymbol{\nabla}_v\varphi(\boldsymbol{x},\boldsymbol{v})\delta(\boldsymbol{x}-\boldsymbol{X}_p)\delta(\boldsymbol{v}-\boldsymbol{V}_p)\ \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{v}\\
&= \int \boldsymbol{\nabla}_x\varphi\cdot(\boldsymbol{v}f) + \boldsymbol{\nabla}_v\varphi\cdot(\boldsymbol{a}f)\ \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{v}\\
&\Rightarrow \int \left[\partial_t f + \boldsymbol{\nabla}_x\cdot(vf) + \boldsymbol{\nabla}_v\cdot(af)\right]\varphi\ \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{v} = 0
\end{aligned}$$

where the integrals are taken over the phase-space and $\boldsymbol{A}_p$ is the particle acceleration. Integration by parts has been used for the last line. Because the last integral doesn't depend on φ, it comes:

$$\partial_t f + \boldsymbol{\nabla}_x\cdot(\boldsymbol{v}f) + \boldsymbol{\nabla}_v\cdot(\boldsymbol{a}f) = 0. \tag{4.14}$$

$\boldsymbol{a} = \boldsymbol{F}/m_p$ is introduced to denote the instantaneous acceleration of the particle. Equation (4.14) is a Liouville equation expressing a conservation of the phase space density, whose dimension is $L^{-2d}T^d$. The second term on the right hand side is the streaming of the particles by the flow while the third term is the change in velocity due to the application of the force $\boldsymbol{F}$.

The knowledge of $f(\boldsymbol{x},\boldsymbol{v})$ allows one to evaluate quantities such as density and momentum in the position space. These are obtained by evaluating the first moments of the distribution f:

0^{th}: number density $$n_p(\boldsymbol{x}) = \int_{\mathcal{V}} f(\boldsymbol{x},\boldsymbol{v})\ \mathrm{d}^d v \tag{4.15}$$

1^{st}: momentum $$n_p(\boldsymbol{x})v_{p_i}(\boldsymbol{x}) = \int_{\mathcal{V}} v_i\, f(\boldsymbol{x},\boldsymbol{v})\ \mathrm{d}^d v \tag{4.16}$$

2^{nd}: kinetic energy $$n_p(\boldsymbol{x})e_{ij}(\boldsymbol{x}) = \int_{\mathcal{V}} v_i v_j\, f(\boldsymbol{x},\boldsymbol{v})\ \mathrm{d}^d v \tag{4.17}$$

with $\mathcal{V}$ the velocity domain. $n_p(\boldsymbol{x})$ is the density in terms of number of particles. For a monodisperse non reacting phase as will be our case, we can define a mass density by $\rho(\boldsymbol{x}) = m_p n_p(\boldsymbol{x})$ with a constant m_p.

A model is needed for the different terms appearing in the kinetic equation (4.14). In our case, as we make use of a phase-space only constituted by positions and velocities, only a model for the force has to be provided. In this chapter, we only consider dilute suspensions so that inter-particle collisions are neglected. For such suspensions, $\boldsymbol{F}$ reduces to a Stokes drag $\boldsymbol{F} = \tau_p^{-1}(\boldsymbol{u} - \boldsymbol{v}_p)$ (see Section 4.1.1).

Solving the kinetic equation

Starting from the kinetic equation (4.14), one has several strategies to solve for the distribution f. One is to integrate it over velocities, once it has been multiplied by v^n which yields evolution equations for the n first moments of velocity. Evolution equations for the macroscopic quantities are then recovered. However, it can be shown that the equation for a moment of order n involves the $(n+1)$-th order. This forms a set of infinite equations yielding an unclosed hierarchy, and this is why closure is necessary at some order n. For example, if M^i denotes the moment of order i, being itself a tensor of order i, the following evolution equations are (Fox, 2012):

$$\begin{aligned}
\frac{\partial M^0}{\partial t} + \frac{\partial M^1_i}{\partial x_i} &= 0, \\
\frac{\partial M^1_i}{\partial t} + \frac{\partial M^2_{ij}}{\partial x_j} &= \frac{1}{\tau_p}(M^0 u_i - M^1_i), \\
\frac{\partial M^2_{ij}}{\partial t} + \frac{\partial M^3_{ijk}}{\partial x_k} &= \frac{1}{\tau_p}(M^1_i u_j + u_i M^1_j - 2M^2_{ij}).
\end{aligned} \tag{4.18}$$

This system is said to be *unclosed* because one needs an evolution equation for M^3, which would make appear M^4, etc. A closed formulation for M^3 is thus needed, which may be built by combining the lower order moments. The quality of the closure may itself be optimised and improved by relevant knowledge of the underlying detailed physics.

As already stated, one must take care that the dynamics of large Stokes numbers particles require a description in the $2d$ phase-space $(\boldsymbol{x}, \boldsymbol{v})$ in order to resolve velocity dispersion. To account for this dispersion, some methods involve for example the integration of the moments up to the second order, then a reconstruction of the full distribution $f(\boldsymbol{x}, \boldsymbol{v})$. This distribution is then used to compute the third order moment M_3 to be used in (4.18). This method however requires to assume the functional form of f (see, for instance, Simonin *et al.* (1993)).

In Aguinaga *et al.* (2009), a closed kinetic equation is developed where the interaction between particles and turbulence is modelled through a return-to-equilibrium term similar to the Bhatnagar–Gross–Krook model (Bhatnagar *et al.*, 1954), namely $\Omega_i = -\tau^{-1}(f - f_p)$, where f_p is an equilibrium distribution and τ the relaxation time. Different models exist then for f_p, including a Gaussian pdf. The resulting equation may then be explicitly solved using finite differences (Aguinaga *et al.*, 2009) or via a Lattice Boltzmann scheme (Fede *et al.*, 2015).

The following sections describe the part of this thesis dedicated to this issue. A novel approach to solve the kinetic equation is presented, based on the integration of the distribution f in the full position-velocity phase space. Some of the results presented here may be found in the paper published in *Comptes Rendus de Mécanique* (Laenen *et al.*, 2016). In this paper, the considered kinetic equation differs from (4.14) by an additional diffusive term, i.e it reads:

$$\partial_t f + \boldsymbol{\nabla}_x \cdot (\boldsymbol{v} f) + \boldsymbol{\nabla}_v \cdot (\boldsymbol{a} f) - \kappa_v \nabla_v^2 f = 0. \tag{4.19}$$

4.3 Description of the method

The solutions $f(\boldsymbol{x}, \boldsymbol{v}, t)$ to the Liouville equation (4.14) are defined in the full position-velocity phase-space $\Omega \times \mathbb{R}^d$, where Ω designates a d-dimensional bounded spatial domain. To simulate numerically the dynamics, we divide the phase-space in $(2 \times d)$-dimensional hypercubes. We then approximate $f(\boldsymbol{x}, \boldsymbol{v}, t)$ as a piecewise-constant scalar field on this lattice. Positions are discretised on a uniform grid with spacing Δx in all directions. In principle, f has to be defined for arbitrary large velocities. We however assume that relevant values of $\boldsymbol{v}$ are restricted to a bounded interval $[-V_{\max}, V_{\max}]^d$, where $V_{\max}$ has to be specified from physical arguments based on the forces $\mathcal{F}$ applied on the particles. Velocities are assumed to take N_v^d values, so that the grid spacing reads $\Delta v = 2\, V_{\max}/N_v$. Figure 4.3 illustrates the phase-space discretisation in the one-dimensional case with $N_v = 5$. The various cells in position-velocity contain a given mass of particles. All these particles are assumed to have a position and velocity equal to that at the centre of the cell.

The three phase-space differential operators appearing in equation (4.14), namely the advection, the particle forcing, and the diffusion, are applied one after the other, following an *operator splitting* method (LeVeque, 2002). For the advection step, we use a technique inspired from the Lattice-Boltzmann method (see Succi (2001)). The time stepping is chosen so that a discrete velocity exactly matches a shift in positions by an integer number of grid-points. Namely, we prescribe $\Delta x = \Delta v\, \Delta t$. All the particle phase-space mass located in $[-\Delta v/2, \Delta v/2]$ does not move; that in $[\Delta v/2, 3\Delta v/2]$ is shifted by one spatial gridpoint to the right and that in $[-3\Delta v/2, -\Delta v/2]$ to the left, etc. All the mass is displaced from one cell to another according to its own discrete velocity value. This evolution is sketched by black horizontal arrows in Figure4.3. This specific choice for the time-stepping implies that the advection (in space) is treated exactly for the discrete system. The next steps consist in applying the force acting on the particles and the diffusion. The corresponding terms in equation (4.14) are conservation laws, which suggests using a finite-volume approximation. The time evolutions due to forcing and diffusion are performed successively. In both

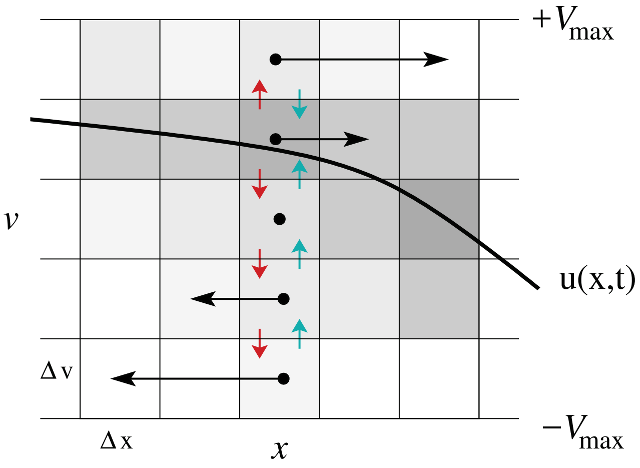


Figure 4.3: Sketch of the lattice dynamics in the (x, v) position-velocity phase space. The solid curve is the fluid velocity profile; the grey-scale tiling represents the discretisation of particles mass in phase space. The black horizontal arrows show advection, while the blue and red vertical arrows are forcing and diffusion, respectively.

cases, we use classical schemes (see below), where zero-flux conditions are imposed on the boundary of $[-V_{\max}, V_{\max}]^d$. The force is evaluated using the values of $\boldsymbol{v}$ at the centres of the cells and $\nabla_v f$ is approximated using finite differences. These steps are illustrated by the horizontal blue and red arrows in Figure 4.3.

A few comments on the convergence and stability of the proposed method. Clearly, except for specific singular forcings, all the linear differential operators involved in (4.14) are expected to be bounded.[1] We can thus invoke the equivalence (or Lax–Richtmyer) theorem for linear differential equations that ensures convergence, provided the scheme is stable and consistent LeVeque (2002).

For the operator associated to particle acceleration, we use in this study either a first-order upwind finite-volume scheme or a higher-order flux limiter by following the strategy proposed in Hundsdorfer *et al.* (1995). The upwind scheme is first-order accurate and is well-known for being consistent and stable if it satisfies the Courant–Friedrichs–Lewy (CFL) condition. This requires that the time needed to accelerate particles by the grid size Δv is larger than the time step Δt, leading to the stability condition

$$\mathrm{CFL} = (\Delta t/\Delta v) \max_{\boldsymbol{x},\boldsymbol{v},t} |\mathcal{F}(\boldsymbol{x}, \boldsymbol{v}, t)|/m_{\mathrm{p}} < 1. \tag{4.20}$$

The upwind scheme is however known to suffer from numerical diffusion, and obviously, one should only expect to recover the correct dynamics only when the numerical diffusion

---

[1] Notice that, although the velocity might explicitly appear in the force $\mathcal{F}$, we only solve for a compact domain of velocities, thus preventing divergences.

κ_{num} is much smaller than the physical one κ. The average numerical diffusion can be estimated as $\langle \kappa_{v,\mathrm{num}} \rangle \approx \langle \mathcal{F} \rangle \, \Delta t / \Delta v$. To limit the effects of this numerical diffusion, we have also used a flux-limiter scheme. While taking benefit of a higher-order approximation where the field is smooth, it uses the ratio between consecutive flux gradients to reduce the order in the presence of strong gradients only. The limiter is a nonlinear function of the phase-space density field and the stability is ensured provided that it is total-variation diminishing (TVD), see LeVeque (2002). Among the various available TVD limiters, we choose the scheme proposed in Koren (1993) with parameter 2/3.

For the term associated to diffusion, the flux at the interface between two velocity cells is computed using finite differences. The resulting finite-volume scheme is thus equivalent to compute a discrete Laplacian on the velocity mesh. The stability condition is then given by

$$\frac{\kappa_v \Delta t}{\Delta v^2} < \frac{1}{2}. \tag{4.21}$$

To summarize, the stability and convergence of the proposed method is ensured when both (4.20) and (4.21) are satisfied.

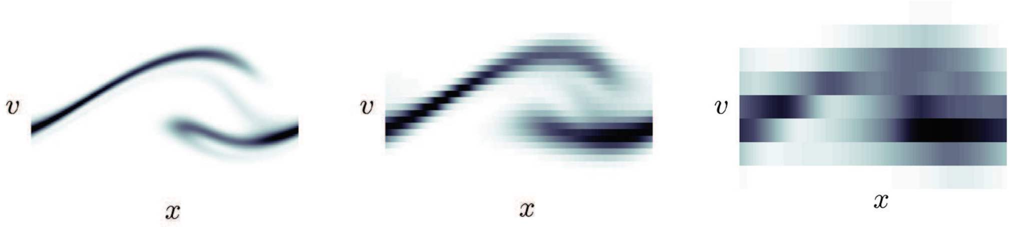
From now on we restrict ourselves to small and heavy particles whose interaction with the carrier fluid is dominated by viscous drag and diffusion. In that case, we have that the acceleration $\boldsymbol{a}_p$ of one particle reads:

$$\boldsymbol{a}_p = \frac{\mathrm{d}\boldsymbol{v}_{\mathrm{p}}}{\mathrm{d}t} = -\frac{1}{\tau_p} \left(\boldsymbol{v}_{\mathrm{p}} - \boldsymbol{u}(\boldsymbol{x}_{\mathrm{p}}, t) \right) + \sqrt{2\kappa_v}\, \boldsymbol{\eta}(t), \tag{4.22}$$

where $\boldsymbol{\eta}(t)$ is the standard d-dimensional white noise and the fluid velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ is prescribed and assumed to be in a (statistically) stationary state. This Stokes drag involves the viscous particle response time $\tau_p = 2\rho_{\mathrm{p}} a^2 / (9 \rho_{\mathrm{f}} \nu)$, with a the radius of the particles, ν the viscosity of the fluid, $\rho_{\mathrm{p}} \gg \rho_{\mathrm{f}}$ the particle and fluid mass densities, respectively. Inertia is quantified by the Stokes number $St = \tau_p / \tau_{\mathrm{f}}$, where τ_{f} is a characteristic time of the carrier flow. The diffusion results from the random collisions between the considered macroscopic particle and the molecules of the underlying gas. Assuming thermodynamic equilibrium, the diffusion coefficient reads $\kappa = 2\, k_{\mathrm{B}}\, T / (m_{\mathrm{p}}\, \tau_{\mathrm{p}})$, where k_{B} is the Boltzmann constant and T the absolute temperature. The effect of diffusion is measured by the non-dimensional number $\mathcal{K} = \kappa\, \tau_{\mathrm{f}} / U_{\mathrm{f}}^2$ (U_{f} being a characteristic velocity of the fluid flow).

Such a specific dynamics leads to appropriate estimates for the bound $V_{\max}$ in particle velocity. One can indeed easily check that when $\kappa = 0$, we always have $|\boldsymbol{v}_{\mathrm{p}}| \leq \max_{\boldsymbol{x},t} |\boldsymbol{u}(\boldsymbol{x}, t)|$. In a deterministic fluid flow, as for instance when $\boldsymbol{u}$ is stationary, this gives the natural choice $V_{\max} = \max_{\boldsymbol{x},t} |\boldsymbol{u}(\boldsymbol{x}, t)|$. However, in most situations, the maximal fluid velocity is not known a priori. One then relies on the statistical properties of $\boldsymbol{u}$, as for instance its root-mean square value $u_{\mathrm{rms}} = \langle u_i^2 \rangle^{1/2}$. Usually the one-time, one-point statistics of fluctuating velocity fields (being random or turbulent) are well described by a Gaussian distribution. This ensures that by choosing $V_{\max} = 3\, u_{\mathrm{rms}}$, the probability that a particle has a velocity out of the prescribed bounds is less than 1%. Such estimates are rather rough. In practice, it is known that the typical particle velocity decreases as a function of the Stokes number. It was for instance shown in Abrahamson (1975) that $\langle|\boldsymbol{v}_{\mathrm{p}}|^2\rangle \propto u_{\mathrm{rms}}^2/St$ at very large Stokes numbers. An efficient choice for $V_{\max}$ should account for that.

The equation solved in the velocity space is analogous to an advection equation in position space. In finite volume and finite difference methods, those are known to suffer from numerical diffusion and to introduce a non physical broadening of the solution along the diffusive dimension, i.e. an increase of the variance of the distribution. For example, the simple upwind scheme yields a numerical diffusion term with diffusivity equals to $\kappa_{\mathrm{num}}(v) = (1 - \frac{a(v)\Delta t}{\Delta v})\frac{a(v)\Delta v}{2}$. For more details, see, for instance, Cushman-Roisin & Beckers (2011).



(a) $N_x = 16384,\ N_v = 127$ (b) $N_x = 4096,\ N_v = 33$ (c) $N_x = 1024,\ N_v = 9$

Figure 4.4: Illustration of the effect of numerical diffusion. Phase space resolutions N_x and N_v are increased at constant Δt with no physical diffusion ($\mathcal{K} = 0$) .

The importance of this numerical diffusion can be quantified by the grid Peclet number

$$Pe_{\mathrm{num}} = \frac{\Delta v\, a_{\mathrm{rms}}}{\kappa}. \tag{4.23}$$

It is thus of importance that $\kappa_{\mathrm{num}} \ll \kappa$.

In the next two sections we investigate two different cases: first a one-dimensional random Gaussian carrier flow with a prescribed correlation time and, second, a two-dimensional turbulent carrier flow that is a solution to the forced incompressible Navier-Stokes equations.

4.4 Application to a one-dimensional random flow

4.4.1 Particle dynamics for $d = 1$

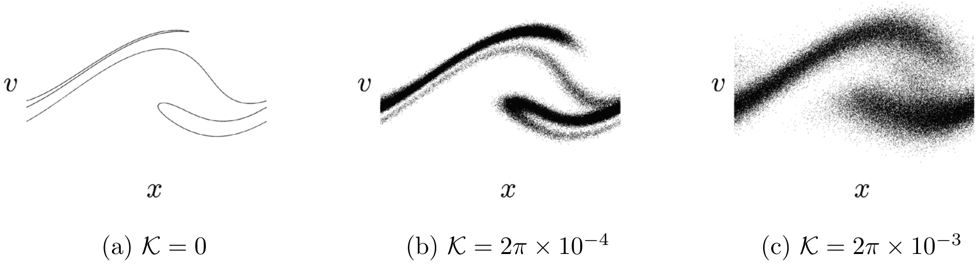
In this section, our method is tested in a one-dimensional situation. For that, we assume that the fluid velocity is a Gaussian random field, which consists in the superposition of two modes whose amplitudes are Ornstein–Uhlenbeck processes, namely

$$u(x,t) = A_1(t)\cos(2\pi\, x/L) + A_2(t)\sin(2\pi\, x/L) \tag{4.24}$$

$$\frac{\mathrm{d}A_i(t)}{\mathrm{d}t} = -\frac{1}{\tau_\mathrm{f}} A_i(t) + \sqrt{\frac{2}{\tau_\mathrm{f}}}\,\xi_i(t) \tag{4.25}$$

where the ξ_i's are independent white noises with correlations $\langle \xi_i(t)\,\xi_i(t')\rangle = u_\mathrm{rms}^2\,\delta(t-t')$. This flow is by definition fully compressible (potential) and spatially periodic with period L. It is characterized by its amplitude $\langle (u(x,t))^2\rangle^{1/2} = u_\mathrm{rms}$ and its correlation time τ_f, which are fixed parameters. We focus on the case when the Kubo number $Ku = \tau_\mathrm{f}\, u_\mathrm{rms}/L$ is of the order of unity.

We next consider particles suspended in this flow and following the dynamics (4.22). The relevant Stokes number is then defined as $St = \tau_\mathrm{p}\, u_\mathrm{rms}/L$ and the relative impact of diffusion is measured by $\mathcal{K} = \kappa L/u_\mathrm{rms}^3$. When diffusion is neglected ($\mathcal{K} \to 0$), the particles distribute on a dynamical attractor (see Figure 4.5 Left) whose properties depend strongly on St. These strange attractors are typically fractal objects in the phase space and they are characterized by their fractal dimension spectrum (Hentschel & Procaccia, 1983). The locations of particles are obtained by projecting theses sets on the position space and might thus inherit the associated clustering (Bec, 2003). The dimension that is relevant for binary interactions between particles is the correlation dimension $\mathcal{D}_2$, which relates to the probability of having two particles within a given distance, namely



(a) $\mathcal{K} = 0$ (b) $\mathcal{K} = 2\pi \times 10^{-4}$ (c) $\mathcal{K} = 2\pi \times 10^{-3}$

Figure 4.5: Instantaneous snapshots of the particle positions in the (x, v) plane for $St \approx 2$ for varying diffusivities $\mathcal{K}$. The folded structures are spread out by diffusion.

$$p_2^<(r) = \mathbb{P}(|\boldsymbol{X}_p^{(1)}(t) - \boldsymbol{X}_p^{(2)}(t)| < r) \sim (r/L)^{\mathcal{D}_2}, \text{ for } r \ll L, \tag{4.26}$$

where $\boldsymbol{x}_\mathrm{p}^{(1)}$ and $\boldsymbol{x}_\mathrm{p}^{(2)}$ denote the positions of two different particles. Note that as we consider u to be in a statistically stationary state, $p_2^<$ is independent of time. The correlation dimension $\mathcal{D}_2$ varies from $\mathcal{D}_2 = 0$ for a point concentrations to $\mathcal{D}_2 = 1$ for a homogeneous mass distribution . In the example of Figure 4.5 (Left) $\mathcal{D}_2 \approx 0.7$. The variations of $\mathcal{D}_2$ as a function of the Stokes number are displayed in the inset of Figure 4.6. $\mathcal{D}_2$ indeed varies from 0 at small Stokes numbers to values close to one. For $St = 0$, the particles concentrate on a point; their distribution is said to be atomic and $\mathcal{D}_2 = 0$. This is a consequence of the compressibility of the one-dimensional (potential) flow. Actually this behaviour persists for finite Stokes numbers, up to a critical value $St^\star$, as shown in Wilkinson & Mehlig (2003) in the case where $\tau_\mathrm{f} \ll L/u_\mathrm{rms}$ (that is $Ku \to 0$). We observe here $St^\star \approx 0.6$. For $St > St^\star$, the dimension increases and tends to a homogeneous distribution ($\mathcal{D}_2 = 1$) at large particle inertia.

When one has only access to the Eulerian density of particles, the distribution of distances cannot be directly inferred from (4.26). One then relies on the coarse-grained density of particles

$$\rho_r(x,t) = \int_{-r/2}^{r/2} \mathrm{d}x' \int \mathrm{d}v\, f(x+x',v,t). \tag{4.27}$$

It is known that, under some assumptions on the ergodicity of the particle dynamics, the second-order moment of this quantity scales as $\langle \rho_r^2 \rangle \propto r^{\mathcal{D}_2-1}$ (see, e.g., Hentschel & Procaccia (1983)). In one dimension, this second-order moment is exactly the same as the radial distribution function. This quantity will be used in the next sections to address the physical relevance of the lattice-particle method. It is of particular interest when considering collisions between particles. Indeed, as explained for instance in Sundaram & Collins (1997), the ghost-collision approximation leads to write the collision rate between particles as the product of two contributions: one coming from clustering and entailed in the radial distribution function, and another related to the typical velocity differences between particles at a given distance. This second quantity relates to the particle velocity (first-order) structure function

$$S_1(r) = \left\langle |\boldsymbol{V}_p^{(1)} - \boldsymbol{V}_p^{(2)}| \,\middle|\, |\boldsymbol{X}_p^{(1)} - \boldsymbol{X}_p^{(2)}| = r \right\rangle. \tag{4.28}$$

This is the average of the amplitude of the velocity difference between two particles that are at a given distance r. As the probability of distances, this quantity behaves as a power law $S_1(r) \sim r^{\zeta_1}$ for $r \ll L$ (see e.g. Bec *et al.* (2005)). The exponent ζ_1, shown in the inset of Figure 4.6 decreases from 1 at $St = 0$, corresponding to a differentiable particle velocity field, to 0 when $St \to \infty$, which indicates that particle velocity differences become uncorrelated with their distances. Again, when working with the phase-space density one cannot use (4.28) but relies on

$$S_1(r) = \frac{\left\langle \int \mathrm{d}v \int \mathrm{d}v'\, f(x,v)\, f(x+r,v')\, |v-v'| \right\rangle}{\left\langle \int \mathrm{d}v \int \mathrm{d}v'\, f(x,v)\, f(x+r,v') \right\rangle}. \tag{4.29}$$

As the second-order moment of the coarse-grained density, this quantity will also be used as a physical observable for benchmarking the method.

In the above discussion, we have neglected the effects of diffusion. It is for instance expected to alter clustering properties by blurring the particle distribution at small scales.

This is illustrated in Figure 4.5 where one can compare the instantaneous phase-space particle positions in the absence of diffusion (Left) and when it is present (Middle and Right) at the same time and for the same realization of the fluid velocity. At large scales, identical patterns are present, but diffusion acts at small scale and smoothes out the fine fractal structure of the distribution. One can easily estimate the scales at which this crossover occurs. Diffusion is responsible for a dispersion v_d in velocities that can be obtained by balancing Stokes drag and diffusion in the particle dynamics, namely $v_\mathrm{d}^2/\tau_p \approx \kappa$, so that $v_\mathrm{d} \sim \tau_p^{1/2}\kappa^{1/2}$. This dispersion in velocity is responsible for a dispersion in positions on scales of the order of $\ell_d = \tau_p\, v_\mathrm{d} \sim \tau_p^{3/2}\kappa^{1/2} = St^{3/2}\mathcal{K}^{1/2}L$. Hence, when diffusion is small enough and $\ell_\mathrm{d} \ll L$, the spatial distribution of particles is unchanged by diffusion at length scales $r \gg \ell_\mathrm{d}$, and the probability that two particles are at a distance less than r behaves as $p_2^<(r) \sim (r/L)^{\mathcal{D}_2}$. For $r \ll \ell_\mathrm{d}$, diffusion becomes dominant, the particles distribute in a homogeneous manner and $p_2^<(r) \propto r^d$, with $d = 1$ being the space dimension. By continuity at $r = \ell_\mathrm{d}$, we get $p_2^<(r) \sim (\ell_\mathrm{d}/L)^{\mathcal{D}_2}(r/\ell_\mathrm{d})$ at small scales.

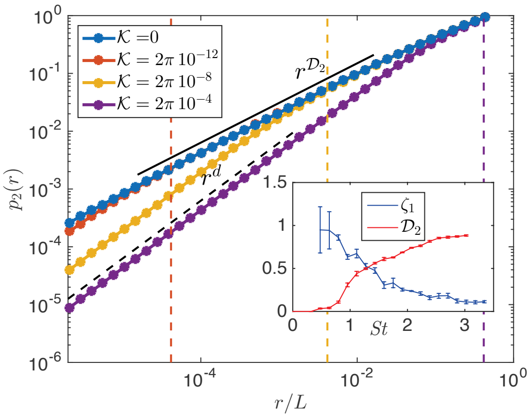


Figure 4.6: Cumulative probability $p_2^<(r)$ of inter-particle distances for various diffusivities κ and for $St \approx 2$. One observes at low diffusivities and for $r > \ell_\mathrm{d}$ a behaviour $\propto (r/L)^{\mathcal{D}_2}$ with $\mathcal{D}_2 \approx 0.7 < d = 1$, followed at small scales by uniform particle distribution where $p_2^<(r) \propto r$. As κ increases, the transition is moved to larger values of r. The vertical dashed represent in each case the estimate $\ell_d \sim \tau_p^{3/2}\kappa^{1/2}$ for this transition. Inset: variations of the correlation dimension $\mathcal{D}_2$ and of the scaling exponent ζ_1 of the first order particle structure function, as a function of the Stokes number St in the case of the random fluid velocity defined by equations (4.24)-(4.25).

This picture is confirmed numerically as shown in Figure 4.6 which represents the scale-behaviour of $p_2^<(r)$ for a fixed Stokes number and various values of the diffusivity κ. One clearly observes the homogeneous distribution $\propto r^d$ at small scales and the fractal scaling $\propto r^{\mathcal{D}_2}$ in an intermediate range. The predicted transition between the two behaviours is indicated by the vertical lines at the diffusive scale ℓ_d. A homogeneous distribution is recovered for $r \lesssim \ell_\mathrm{d}/10$.

Velocity statistics are also altered by the presence of diffusion. The structure function $S_1(r)$ is expected to behave as r^{ζ_1} for $\ell_\mathrm{d} \ll r \ll L$ and to saturate to a constant value when $r \ll \ell_\mathrm{d}$. By continuity, the value of this plateau should be $\sim \ell_\mathrm{d}^{\zeta_1} \sim \mathcal{K}^{\zeta_1/2}$. Note finally that the slow convergence $\ell_\mathrm{d}/L \propto \sqrt{\mathcal{K}}$ as $\mathcal{K} \to 0$ implies that very small values of the diffusion are needed in order to clearly recover the statistics of diffusive-less particles as an intermediate asymptotics.

4.4.2 Lattice-particle simulations

We now turn to the application of the lattice-particle method described in Section 4.3 to this one-dimensional situation. We compare the results to Lagrangian simulations where we track the time evolution of N_p particles randomly seeded in space with zero initial velocity. We choose and normalize the initial phase-space density $f(x, v, 0)$ to match the Lagrangian settings. The distribution is uniform over the cells, concentrated on a vanishing velocity and the total mass is such that $\sum_{i,j} f(x_i, v_j, t)\, \Delta x\, \Delta v = N_p$. In all simulations, the maximum velocity is set to $V_\mathrm{max} = u_\mathrm{rms} = 1$ and we have chosen $L = 2\pi$ and $\tau_\mathrm{f} = 1$. In these units, the time step is kept fixed at $\Delta t = \Delta x/\Delta v = 2^{-6}\pi \approx 0.05$. The number of discrete velocities is of the form $2^n + 1$ and is varied between $N_v = 3$ to 129. The number of spatial collocation points is then given by 2^{n+6} and thus varies between $N_x = 128$ to 8192. Note that, because of the CFL condition (4.20), this choice restricts the number of discrete velocities that can be used to $N_v < 1 + 128\, St/(2\pi)$.

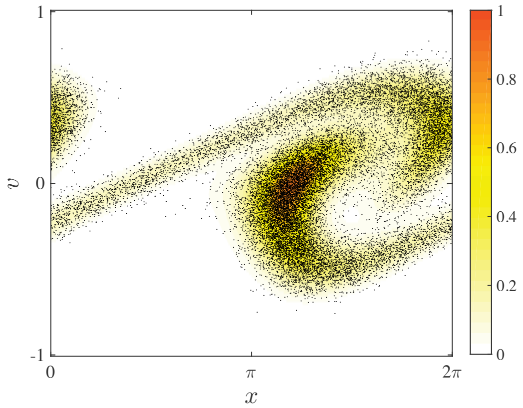


Figure 4.7: Position-velocity phase-space positions of Lagrangian particles (black dots) on the top of the field obtained by the lattice-particle method (coloured background). The diffusivity is here $\kappa = 10^{-3}$ ($\mathcal{K} \approx 6.28.10^{-3}$) and $St \approx 2$.

Figure 4.7 represents simultaneously the phase-space distribution of Lagrangian particles and the numerical approximation obtained by the particle-lattice method for $N_v = 129$. Clearly, one observes that the method fairly reproduces the distribution of particles, including the depleted zones, as well as the more concentrated regions. Furthermore, the method is able to catch multivalued particle velocities. We have for instance up to three branches in v for $x \simeq 3\pi/2$. It is important to emphasize that numerical diffusion is of course present, and that it has to be smaller than the physical diffusion κ in order for the approach to be consistent with the Lagrangian dynamics.

To get a more quantitative insight on the convergence of the method, we next compare the coarse-grained densities obtained from the Lagrangian simulation and the lattice-particle approximation of the phase-space density. The first, denoted ρ_r^{L} is computed by counting the number of particles contained in the different boxes of a tiling of size r. The second is written as ρ_r^{E} and is obtained by summing over velocities and coarse-graining over a scale r the phase-space density obtained numerically. To confirm the convergence of the method, we measure for a fixed r the behaviour of the $\mathcal{L}_2$-norm of the difference between ρ_r^{L} and ρ_r^{E}, namely

$$\|\rho_r^{\mathrm{L}} - \rho_r^{\mathrm{E}}\| = \left\langle \left(\rho_r^{\mathrm{L}}(x,t) - \rho_r^{\mathrm{E}}(x,t)\right)^2 \right\rangle^{1/2}, \tag{4.30}$$

where the angular brackets $\langle\cdot\rangle$ encompass a spatial and a time average. Figure 4.8 shows

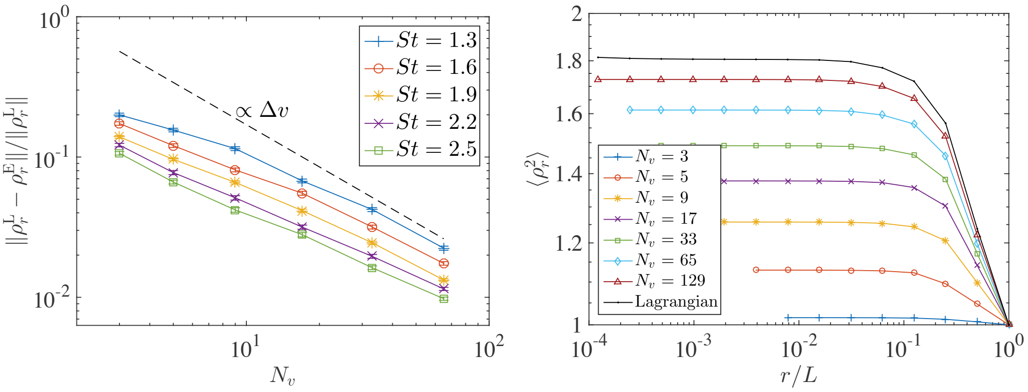


Figure 4.8: Left: Relative $\mathcal{L}_2$-error of the lattice-particle method for evaluating the coarse-grained density ρ_r over a scale $r = L/128$ as a function of the number of velocity grid-points N_v and for various values of the Stokes number, as labelled. Right: Convergence of the second-order moment of the coarse-grained density $\langle\rho_r^2\rangle$, which is shown as a function of r for $St \approx 1.9$, $\mathcal{K} = \pi\, 10^{-2}$, and various lattice velocity resolutions N_v, as labelled.

the behaviour of the relative $\mathcal{L}_2$-error as a function of the number of velocity grid-points N_v, for various values of the Stokes number St and for a given scale r. One observes that the error decreases when the resolution increases, giving strong evidence of the convergence of the method. The error is found proportional to the velocity grid spacing Δv, indicating that the method is first order for large values of N_v. The constant is a decreasing function of the Stokes number. This indicates that the method is more accurate for particles with strong inertia. The reason for this trend will be addressed in the sequel.

To assess the ability of the proposed method to reproduce physically relevant quantities, we now compare statistics obtained using the lattice method with those using a Lagrangian approach. We focus on the clustering and velocity difference properties that were introduced and discussed in section 4.4.1.

Figure 4.8 shows for given values of the Stokes number and of the diffusivity, the second-order moment of the coarse-grained density $\langle(\rho_r^{\mathrm{E}})^2\rangle$ as a function of r and various values of the resolution in velocities, together with the value $\langle(\rho_r^{\mathrm{L}})^2\rangle$ obtained with 10^6 Lagrangian particles. One observes that the curves approach the limiting behaviour from below when the number of grid-points N_v becomes larger (i.e. when $\Delta v \to 0$). At sufficiently high velocity resolutions, the method is able to capture the large-scale properties of the concentration of the particles. The second-order moment of density then saturates to a value lower than that expected from Lagrangian measurements. The situation is very different at very low resolutions where the data obtained from the lattice-particle method deviates much, even at large scales. This corresponds to the case when the numerical diffusion in velocity is larger than the physical diffusion.

These strong deviations stem from a non-trivial effect of diffusion that lead to finite-scale divergences of the solutions associated to different values of $\mathcal{K}$. In the absence of diffusion, there is a finite probability that an order-one fraction of mass gets concentrated on an arbitrary small subdomain of the position-velocity phase space. This corresponds to a violent fluctuation where the local dimension approaches zero. At the time when this occurs, the mass distribution associated to a finite value of the diffusion will get stacked at a scale ℓ_{d}. Because of the chaotic nature of the particle dynamics, the two mass distributions, with and without diffusion will experience very different evolutions and diverge exponentially fast. Such a strong clustering event followed by the divergence of the solutions, is shown in Figure 4.9. Starting from a correctly reproduced distribution, the major part of non-diffusive Lagrangian particles concentrate into a subgrid region while the Eulerian approximation is stacked at scales of the order of ℓ_{d}. At a later time, the two distributions diverge and the diffusive particles fill faster larger scales. The probability with which one encounters such a configuration strongly depends on the Stokes number and on the spatial dimension. In the one-dimensional case, such events are rather frequent but become sparser when the Stokes number increases. This is essentially due to the compressibility of the carrier flow. For incompressible fluids in higher dimensions, we expect a negligible contribution from these events.

In addition, we report some results on velocity difference statistics. For that, we have measured the first-order structure function $S_1(r)$ of the particle velocity, using (4.28) in the Lagrangian case and (4.29) for solutions obtained with the lattice-particle method. Figure 4.10 shows the relative error of $S_1(r)$ for fixed values of the separation $r = 2\pi/2^6 = 2^5\Delta x$, the Stokes number, and the diffusivity, as a function of the velocity resolution.

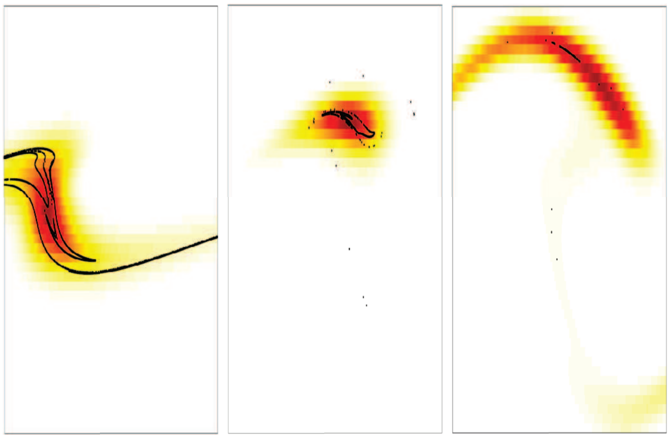


Figure 4.9: Three snapshots of the Lagrangian particles (black dots, for $\mathcal{K} = 0$) and of the lattice-particle Eulerian solution (coloured background) in the (x, v) plane for different times: At $t < t_\star$ (Left) the solution is well approximated at large scales; At $t = t_\star$ (Centre) an order-unity fraction of the mass is concentrated on a scale less than ℓ_d; At $t > t_\star$ (Right), the Eulerian and Lagrangian solutions diverge exponentially fast with differences appearing at the largest scales..

Clearly, when the number of grid-points N_v increases, the error decreases, following a law approximatively proportional to the grid spacing Δv. The inset shows the same quantity but, this time, for a fixed resolution ($N_v = 33$) and as a function of the Stokes number. One clearly observes a trend for this error to decrease with St. There are two explanations for this behaviour. First, as seen above, there are strong clustering events leading to differences between the Lagrangian and lattice solutions that can persist for a finite time. When the Stokes number increases, such events become less probable. The second explanation relies on the fact that particles with a larger Stokes number experience weaker velocity fluctuations. This implies that for a fixed value of V_max, the particle velocity is more likely to be fully resolved at large values of St. As seen in the inset of Figure 4.10, the downtrend of the error is compatible with a behaviour $\propto St^{-1/2}$. It might thus be proportional to the expected value of the root-mean-squared particle velocity when $St \gg 1$ (see Abrahamson (1975)), favouring the second explanation.

To close this section on one-dimensional benchmarks of the lattice-particle method, we briefly assess the numerical cost of the method. The computational cost per time step for both Lagrangian and Lattice simulations are compared for a fixed resolution and coarse-graining scale. A reference coarse-grained density field ρ_r^∞ from the Lagrangian method using a large number of particles ($N_p = 10^7$) is compared to the ones ρ_r^NUM obtained either from the Lattice or the Lagrangian simulations varying the resolution (number of velocities

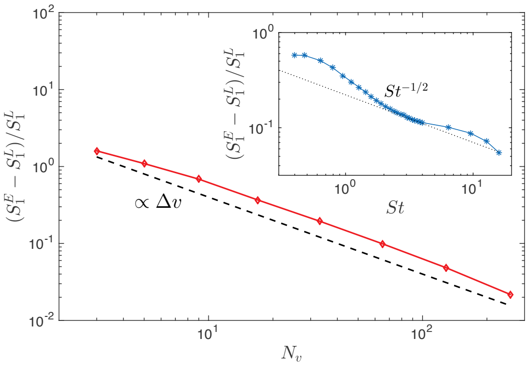


Figure 4.10: Relative error between the particle velocity structure function $S_1^E(r)$ obtained from the lattice-particle method and that $S_1^L(r)$ from Lagrangian averages, as a function of the number N_v of velocity grid-points. Here, the Stokes number is fixed $St \approx 1.9$ and $\mathcal{K} = 2\pi\,10^{-3}$. Inset: same quantity but for $N_v = 33$ and as a function of the Stokes number St.

N_v for the lattice method or the number N_p of Lagrangian particles). All the density fields are coarse-grained at the same scale $r = L/128$ as in Figure 4.8. Figure 4.11 shows the relative error as a function of the computational cost. All the simulations were done using the same quadri-core CPU with shared memory. We expect the costs to have the same tendencies when using distributed memory machines, as the communication should not vary much from one method to the other. For the Lagrangian simulations, the error is essentially given by finite number effects that affect the resolution of the p[article density field. It is thus related to the statistical convergence of the average and behaves as $N_p^{-1/2}$. As the computational cost is proportional to N_p, this explains the 1/2-scaling observed in Figure 4.11. For the lattice simulations the cost is proportional to N_v^2, since increasing N_v with constant time step implies increasing N_x by the same factor. It appears that the Lattice method is more computationally efficient down to a given error. A few number of discrete velocities reproduces indeed better the reference density field than Lagrangian simulations with too few particles. The crossover observed in Figure 4.11 is due to the scaling of the error with respect to N_v that is slightly slower than linear, as shown in Figure 4.8. Note that if the advection term in equation (4.14) is treated with a higher order scheme, the error as a function of N_v will decrease faster than linearly, and the Lattice method will be computationally advantageous if lesser errors are targeted. Finally, let us stress that the comparison is here performed for the simple case of non interacting particles. In some situations involving long-range interactions (such as gravitational or electrical forces), Lagrangian methods might require $\mathrm{O}(N_p^2)$ operations and become much less efficient that the lattice method, which will then still have a cost $\mathrm{O}(N_x \times N_v)$.

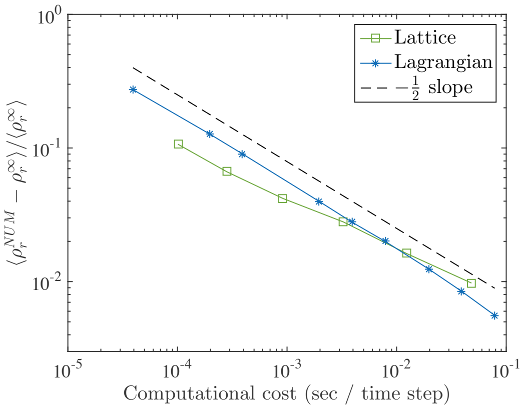


Figure 4.11: Relative $\mathcal{L}_2$ errors on the coarse-grained density fields obtained with either Lagrangian or lattice simulations, plotted here as a function of the computational cost (same parameters as in Figure 4.8 for $St = 2.5$). The various simulations were done with a fixed time stepping $\Delta t = 5 \times 10^{-2}$; the lattice case spans $N_v = 3$ to 65, while the Lagrangian simulations correspond to a number of particles N_p varying from 10^3 to 2×10^6. The symbol in the y-axis label stands for either Lagrangian of Eulerian simulations.

4.5 Application to incompressible two-dimensional flows

Numerical integration in more than one dimension consists of the same routines as in Section 4.3, each operator being applied on one dimension after the other.

4.5.1 Cellular flow

We first consider a fluid flow that is a stationary solution to the incompressible Euler equations (and to the forced Navier–Stokes equations). It consists of a cellular flow field, a model that have often been used to investigate mixing properties, as well as the settling of heavy inertial particles (see, e.g., Maxey (1987); Bergougnoux *et al.* (2014)). The velocity field is the orthogonal gradient of the L-periodic bimodal stream function $\psi(x, y) = U \sin(\pi(x + y)/L) \sin(\pi(x - y)/L)$ (the typical velocity strength is here denoted by U). The cellular flow has been here tilted by an angle $\pi/4$ in order to avoid any alignment of the separatrices between cells with the lattice that leads to spurious anisotropic effects.

Figure 4.12 shows two snapshots for two different values of $St = \tau_p U/L$ of the stationary particle distribution (black dots), together with the density field evolved by the lattice-particle dynamics. For the smallest Stokes number (Left panel), one observes that the particle distribution is concentrated along the separatrices between the different cells. One also observes that it develops entangled structures in the vicinity of the hyperbolic

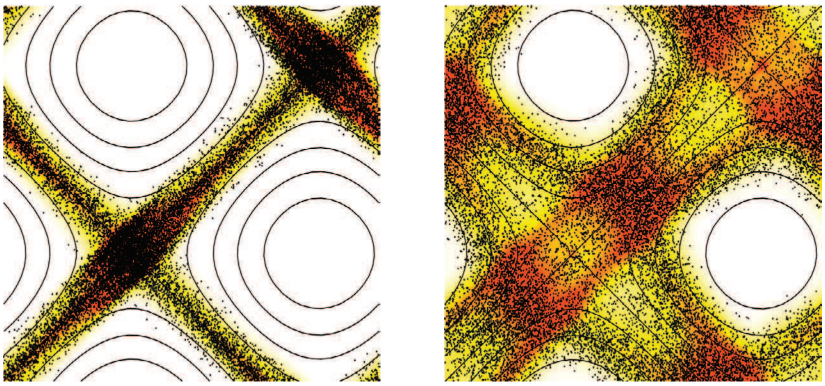


Figure 4.12: Particles stationary distribution inside a tilted cellular flow along with the density field from the lattice method. The value of the diffusivity is $\mathcal{K} = 8\pi \times 10^{-3}$. Left: $St = 1/(2\pi)$. Right: $St = 1/\pi$. These simulations were performed on a lattice with 1024^2 spatial grid-points associated to 19^2 discrete velocities. The distributions have been here spatially shifted in order to avoid having the concentration point $(0, 0)$ at the origin.

stagnation points of the flow. These loops, which are aligned with the stable direction, corresponds to oscillations in the particle dynamics that occurs when their inertia makes them cross the unstable manifold with a too large velocity. At larger St, the particle distribution is somewhat broader but is this time centred on specific trajectories that do not perform the aforementioned oscillation but rather cross ballistically the heteroclinic separatrices. In both cases the particle distribution contains regions where trajectories are clearly crossing each other. The lattice method reproduces fairly well this complex dynamics. Note that any traditional Eulerian-Eulerian method introducing a particle velocity field will not be able to reproduce such effects.

One may wonder why the flow has been tilted by an angle. Figure 4.13 (left) shows the density field obtained for a simulation with a non-tilted flow for $St = 1/\pi$. Unfortunately, it is unable to recover the correct particle spatial repartition, shown on the right panel by the Lagrangian particles (black dots). Actually, mass has concentrated along the steady, stable manifolds between the rotating cells. At each passage through the hyperbolic points (cell corners), for which $\boldsymbol{u} = 0$, a little amount of mass is transferred into the corresponding cell in velocity space $[-\Delta v/2, \Delta v/2]^2$, from which no mass ever escape, due to the stationarity of the velocity field. Trying to circumvent this effect, a refinement of the finite volume scheme was attempted, adding an outgoing flux proportional to the difference between $\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{v}_i$, with $\boldsymbol{v}_i$ the centre of the cell. This ensured an ejection even for the cell for which $\boldsymbol{v}_i = \boldsymbol{0}$. But it only spread the distribution a little along the concentrating, stable manifold. This effect is expected to be present for all finite Δv. This presently limits the capacity of the model to represent correctly flows with stationary manifolds aligned with

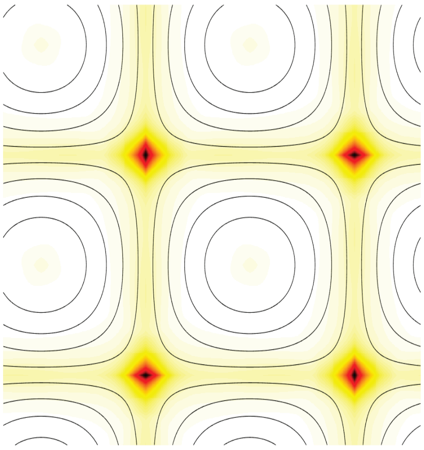


Figure 4.13: Illustration of the particle density integrated by the lattice method in the case of a non-tilted cellular flow. $St = 1/\pi$ and $\mathcal{K} = 8\pi \times 10^{-3}$. This picture is to be compared with Figure 4.12 (right).

one of the lattice propagation direction.

4.5.2 Heavy particles in 2D turbulence

We next turn to the study of the model in non-stationary fluid flows that are solutions to the forced two-dimensional incompressible Navier–Stokes equations (2.1).

The fluid velocity field $\boldsymbol{u}$ is computed numerically using a pseudo-spectral, fully dealiased GPU solver for the vorticity streamfunction formulation of the Navier–Stokes equation (2.18).

The two-dimensional Navier–Stokes equation is known to develop two cascades, as explained in Section 2.2.2.

Dimensional analysis predicts that the direct enstrophy cascade is associated to a unique timescale $\tau_\Omega = \epsilon_\omega^{-1/3}$. Investigating heavy particle dynamics at the small scales of two-dimensional turbulence thus requires comparing their response time to τ_Ω. The relevant parameter is then the Stokes number defined as $St = \tau_p/\tau_\Omega$. For $St \ll 1$, particles almost follow the flow and tend to distribute homogeneously in space. When $St \gg 1$, they completely detach from the fluid and experience a ballistic motion leading again to a space-filling distribution. Non-trivial clustering effects occur when the Stokes number is order one. This is illustrated in Figure 4.14, which shows a snapshot of the particle distribution in the position space on top of the turbulent vorticity field in the direct enstrophy cascade. Due to their inertia, particles are ejected from vortices and concentrate in high-strain regions.

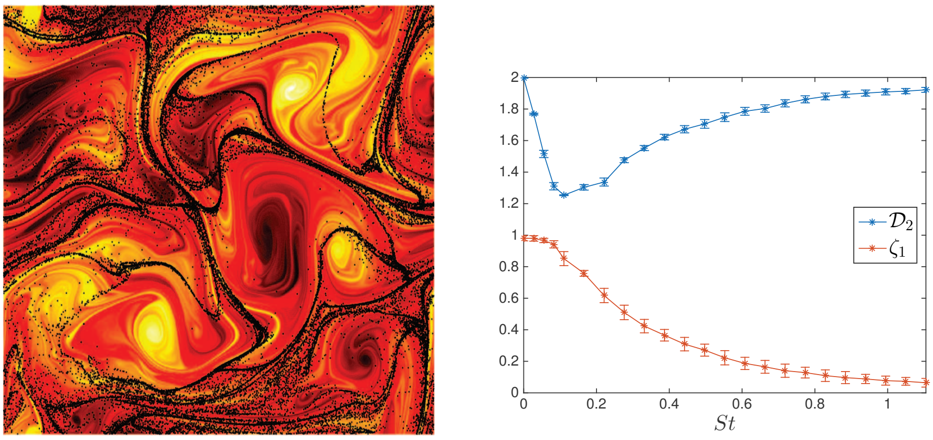


Figure 4.14: Left: Snapshot of the position of particles (black dots) for $St = 0.1$. The coloured background shows the vorticity field obtained from a 1024^2 direct numerical simulation with a large-scale forcing at wavenumbers $1 \leq |\boldsymbol{k}| < 4$. Right: Correlation dimension $\mathcal{D}_2$ and scaling exponent ζ_1 of the particle velocity first-order structure function as a function of St in the two-dimensional direct enstrophy cascade.

There, the combination of stretching, folding and dissipation induced by their dynamics makes them converge to a dynamical attractor with fractal properties. Such a behaviour is quantitatively measured by the correlation dimension $\mathcal{D}_2$ defined in equation (4.26). The evaluation of $\mathcal{D}_2$ as a function of St resulting from Lagrangian simulations is presented in Figure 4.14. At $St = 0$, unlike in the one-dimensional case where the dimension of the attractor is 0, particles follow the streamlines of the incompressible two-dimensional flow, fill the position space, and hence $\mathcal{D}_2 = 2$. Clustering then increases with inertia to attain a minimum at $St \approx 0.2$. It then decreases again as the velocity of particles separate from that of the fluid and disperse in the velocity space, leading to a space-filling distribution $\mathcal{D}_2 = 2$ when $St \to \infty$.

The velocity distribution of particles is itself having a behaviour that is very similar to the one-dimensional case. This is clear from Figure 4.14, where the scaling exponent ζ_1 of its first-order structure function (see equation (4.28)) is represented as a function of the Stokes number. For $St \ll 1$, the particles are as if advected by a smooth velocity field and $\zeta_1 \approx 1$. When $St \gtrsim 1$, particles with very different velocities can come arbitrarily close to each other and $\zeta_1 \to 0$.

Particle properties in the inverse energy cascade are more difficult to characterize because of the scale-invariance of the velocity field. In particular, neither the moments of the coarse-grained density nor the particle velocity structure functions display any scaling behaviour. What has been nevertheless observed numerically by Boffetta *et al.* (2004) is that the particle spatial distribution is dominated by the presence of voids whose sizes obey a universal scaling law. Chen *et al.* (2006*a*) argued that such voids are related to the excited regions of the flow and that particles tend to follow the calm regions where the zeros of the fluid acceleration are more probable.

In the sequel we apply the lattice method to both the direct and the inverse two-dimensional cascades. Resolving both cascades in the same simulation would require a tremendous scale separation and thus number of grid-points (see Boffetta & Musacchio (2010)). For that reason we consider the two cases separately.

Direct enstrophy cascade

The fluid flow is integrated by a pseudo-spectral method on a uniform square spatial grid using a streamfunction formulation of the Navier–Stokes equation (2.18). To maintain a developed turbulent state, a stochastic forcing is applied in the wavenumber shell $1 \leq |\boldsymbol{k}| < 4$ of Fourier space while the kinetic energy accumulating at large scales is removed by a linear friction. The particle dynamics is simulated using a spatial lattice with the same resolution as the fluid and with various numbers N_v^2 of discrete velocities. The acceleration step is done via a flux limiter scheme as described in Section 4.3. Results are compared to particle trajectories obtained from Lagrangian simulations. Figure 4.15 shows the instantaneous particle distributions obtained from the two approaches. The qualitative agreement is excellent, reproducing correctly depleted zone as well as more concentrated regions.

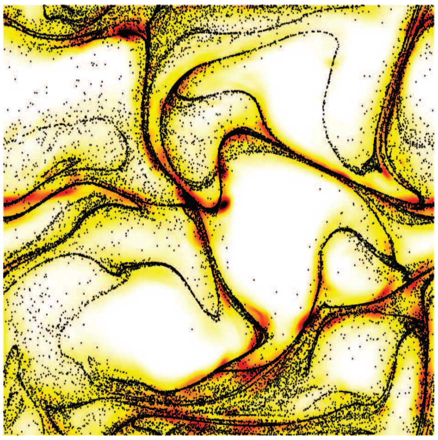


Figure 4.15: Snapshot of the position of Lagrangian particles (black dots), together with the density field obtained from the lattice method (coloured background, from white: low densities to red: high densities) for $St \approx 0.1$ at the same instant of time as Figure 4.14. The fluid flow was integrated using a resolution of 1024^2 and lattice simulations were performed with 1024^2 spatial grid-points associated to 17^2 discrete velocities.

To get more quantitative informations on the relevance of the method, we have performed a set of simulations with a 512^2 resolution and in which both the number of discrete velocities N_v^2 and the maximum velocity $V_{\max}$ are varied. Figure 4.16 shows measurements of the second-order moment of the coarse-grained density ρ_r obtained by integrating the phase-space density $f(\boldsymbol{x}, \boldsymbol{v}, t)$ with respect to velocities and over space in boxes of length r. This is the two-dimensional version of Figure 4.8 and the statistics of $\langle\rho_r^2\rangle$ have a very similar behaviour as in the one-dimensional case. Here $St = 0.5$, $V_{\max} = 3.9\,u_{\rm rms}$, and N_v is varied from 9 to 21. One clearly observes that the statistics obtained from the lattice method converges to that obtained from Lagrangian simulations.

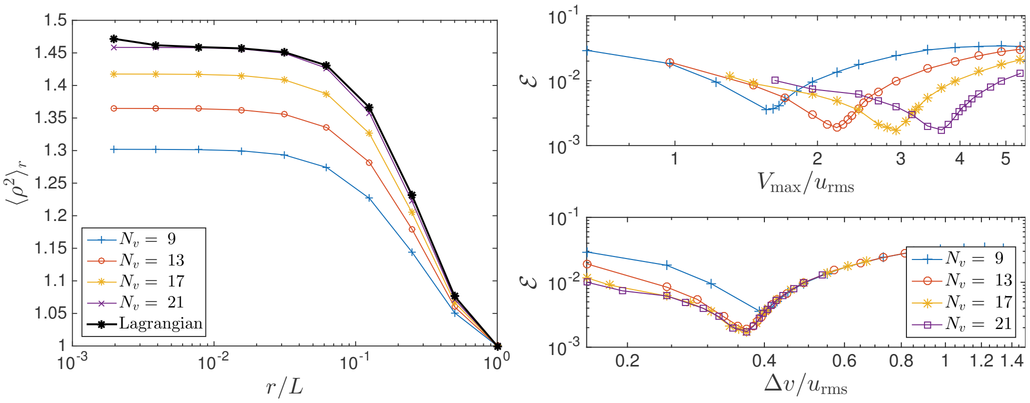


Figure 4.16: Left: Second-order moment of the coarse-grained density ρ_r as a function of r for $St = 0.5$ and $\mathcal{K} = 1.6\pi\ 10^{-1}$ in the direct cascade, both from Lagrangian measurement (black line) and the lattice method with different N_v, as labeled. Right: Distance-averaged error $\mathcal{E}$ of the second-moment of the mass density as defined in (4.31) as a function of the maximal velocity $V_{\max}$ (top) and of the velocity grid spacing $\Delta v = 2\,V_{\max}/(N_v - 1)$ (bottom) for various values of the velocity resolution N_v^2 and for $St \approx 0.5$.

The interplay between the choices of N_v and of $V_{\max}$ requires some further comments. On the one hand the method converges when both $\Delta v = 2\,V_{\max}/N_v \to 0$ and $V_{\max} \to \infty$. On the other hand, the computational cost is $\propto N_v^2$. One can thus wonder if for a fixed cost there is an optimal choice of $V_{\max}$ that minimizes the error obtained with the lattice method. Focusing again on the second-order statistics of the particle mass distribution, we have measured the average with respect to r of the error made on the density moment $\langle\rho_r^2\rangle$ defined as

$$\mathcal{E}(N_v, V_{\max}) = \overline{\langle(\rho_r^{\rm E})^2\rangle - \langle(\rho_r^{\rm L})^2\rangle}/\overline{\langle(\rho_r^{\rm L})^2\rangle}, \tag{4.31}$$

where

$$\overline{f(r)} = \frac{1}{L^2}\int_0^L |f(r)|\, r\, \mathrm{d}r. \tag{4.32}$$

Figure 4.16 (top) represents this quantity as a function of $V_{\max}$ for different values of the cost N_v^2. One clearly observes that there is indeed for a fixed N_v a specific choice of $V_{\max}$ where the error is minimal. The optimal value of the maximal velocity increases with N_v. On the right of the minimum, the error is in principle dominated by a Δv too large. This is confirmed by the collapse of the various curves on the right of their minima that can be seen in the bottom of Figure 4.16 where $\mathcal{E}$ is represented as a function of Δv. In the left of the minimum, the error should be dominated by a too small value of $V_{\max}$. One can indeed guess an asymptotic collapse for $V_{\max} \ll u_{\rm rms}$ on the upper panel of Figure 4.16, or equivalently, the fact that the curves separate from each other at small values of Δv in the lower panel.

The value of the error at the optimal $V_{\max}$ decreases from $N_v = 9$ to $N_v = 13$ but then seems to saturate (or to decrease only very slowly) at higher values of N_v. One cannot exclude that this behaviour corresponds to a logarithmic convergence of $\mathcal{E}$ when $N_v \to \infty$. This slow dependence is also visible in the bottom panel where the collapse of the various curves seems to extend weakly on the left-hand side of the minima for $N_v = 13$, 17 and 21. Accordingly, a small difference in N_v is not enough to decrease significantly the error. In the specific case considered (for $St \approx 0.5$ in the direct cascade), the resulting optimal choice seems to be $N_v = 13$ with $V_{\max} = 2.25\, u_{\rm rms}$, which leads to a relative error 10^{-3}.

Inverse energy cascade

To complete this study we have also tested the proposed lattice method in a two-dimensional turbulent flow in the inverse kinetic energy cascade regime. The stochastic forcing is now acting at small scales ($400 \leq |\boldsymbol{k}| \leq 405$) and we made use of hyper-viscosity (here eighth power of the Laplacian) in order to truncate the direct enstrophy cascade. The kinetic energy accumulated at large scales is again removed using a linear friction in the Navier–Stokes equation (2.18). The particle Stokes number is now defined as $St = \tau_{\rm p}/\tau_L$ using the large-eddy turnover time $\tau_L = L/u_{\rm rms}$ since small-scale statistics are dominated by forcing and are thus irrelevant. The flow is integrated with a resolution of 2048^2 grid-points while the lattice-particle method is applied for $St \approx 0.1$ on a coarser grid with 512^2 points.

Figure 4.17 shows that the lattice-particle method is able to reproduce the main qualitative features of the particle spatial distribution at scales within the inertial range of the inverse energy cascade. This is confirmed in Figure 4.17 which represents the relative error $\mathcal{E}$ defined in equation (4.31) of the second-order moment $\langle \rho_r^2 \rangle$ of the density ρ_r coarse-grained over a scale r. The Lagrangian integration was performed with 2×10^7 particles (with no physical diffusion) and the lattice method on a 512^2 spatial grid with $N_v^2 = 9^2$ discrete values of the particle velocity. One clearly observes that the error decreases at the largest scales of the flow.

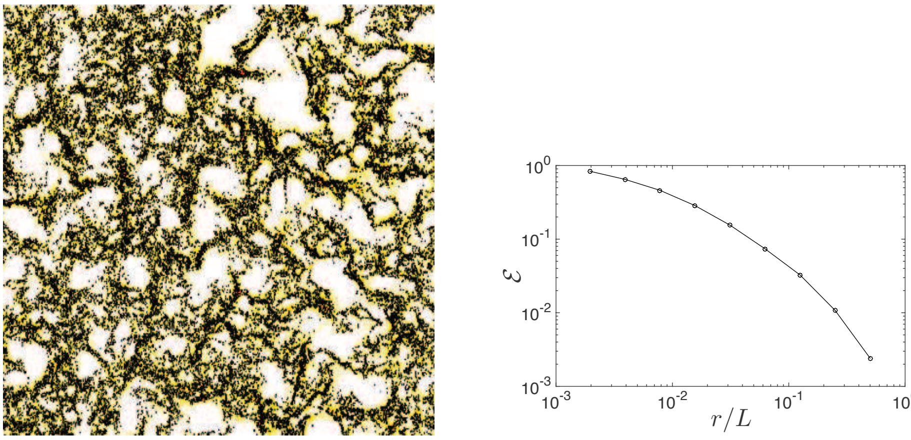


Figure 4.17: Left: Snapshot of the position of Lagrangian particles (black dots), together with the density field obtained from the lattice method (coloured background, from white: low densities to red: high densities) for $St \approx 0.1$ in the inverse energy cascade. The fluid flow was integrated using a resolution of 2048^2 and lattice simulations were performed with 512^2 spatial grid-points associated to 9^2 discrete velocities. Right: Relative error $\mathcal{E}$ of the second-order moment of the coarse-grained density ρ_r as a function of r for $St = 0.5$. The lattice-particle method was here used with 512^2 position grid-points and 9^2 velocity grid-points.

4.6 Conclusions

We have presented a new Eulerian numerical method to simulate the dynamics of inertial particles suspended in unsteady flow. This lattice-particle method is based on the discretisation in the position-velocity phase space of the evolution equation for the particle distribution. The spatial grid is chosen such that particles with a given discrete velocity hop by an integer number of gridpoints during one time step, an idea close to that used in lattice-Boltzmann schemes. We have shown that the method reproduces the correct dynamical and statistical properties of the particles, even with a reasonably small amount of velocity gridpoints. Some deviations from Lagrangian measurements are nevertheless observed at small scales in one dimension. We obtained evidence that they are due to numerical diffusivity acting in the space of velocities and are more important in one dimension at small Stokes numbers than otherwise. The proposed method is anyway intended to describe large scales where such deviations disappear. It might hence be a suitable candidate for developing large-eddy models for particle dynamics. Indeed, as equation (4.19) is linear in f, some techniques of subgrid modeling used in scalar turbulent transport (see Girimaji & Zhou (1996) for example) could be adapted.

Our approach consists in always imposing the same mesh for particle velocities, independently of the spatial position and of the local value of the fluid velocity. This is particularly well-adapted for particles with a large Stokes number. Their velocity experiences small fluctuations and is generally poorly connected to that of the fluid. In addition, the method is accurate at the largest scales and can hence catch the structures appearing in the spatial and velocity distributions of large-Stokes-number particles. Such considerations indicate that the proposed lattice-particle method is suitable for simulating particles with a sufficiently strong inertia. Conversely, particles with a weak inertia develop fine-scale structures in their distribution. They result from tiny departures of their velocity from that of the fluid. Our method, applied with a fixed velocity resolution, might not be able to catch such deviations. However, a more suitable idea for this case is to use a variation of our approach where, instead of a full resolution of the particle velocity, one considers its difference with that of the fluid. This would of course require changing scheme for integrating advection but should in principle not lead to any major difficulty.

CHAPTER 5

Turbulence modulation by small heavy particles

In this chapter, turbulence modulation by small heavy particles is studied in the two-dimensional enstrophy cascade. The two asymptotics of small and high inertia are considered. In both of them, we make use of a fluid description for the solid suspension. This allows us to treat two-way coupling in an exact manner. The net force exerted by the particles on the fluid phase is computed exactly as the result of the action-reaction rule from the number density function.

In the small Stokes regime, the bi-fluid approximation is used, yielding an evolution equation for the density and the velocity of the solid phase. It is shown that particles ejected from vortices core develop a shear instability at their boundaries. This results in a decrease of the slope of the fluid kinetic energy spectrum, together with an enstrophy injection. This last effects results in the reversal of the sign of the nonlinear flux.

This study has made the object of a paper submitted to the journal *Physical Review Letters* which is reproduced hereafter.

In the large Stokes asymptotics, the lattice particle method presented in the previous chapter was coupled to the fluid phase through the particle number density function. The effect of two-way coupling in this case was shown to behave as an additional effective large-scale friction. As the inertia was decreased, the qualitative behaviour observed in the limits of small Stokes numbers (see above) began to be recovered.

This study has made the object of a paper submitted to the journal *Journal of Fluid Mechanics* which is reproduced hereafter.

Modulation of two-dimensional turbulence by small heavy particles

François Laenen,[1] Stefano Musacchio,[2] and Jérémie Bec[1]

[1]Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire J.-L. Lagrange, 06300 Nice, France.

[2]Université Côte d'Azur, CNRS, Laboratoire J.-A. Dieudonné, 06108 Nice, France

(Received xx; revised xx; accepted xx)

The question of two-way coupling, that is of the back effects of a transported particle phase onto the fluid flow, is adressed in this paper. The suspension considered is that of small-Reynolds-number heavy particles with Stokes numbers of the order of unity or larger, that are embedded in a two-dimensional turbulent flow in the direct enstrophy cascade. A recently developed numerical approach providing a description of the particle in terms of fields is used. This method is based on the kinetic equation for the position-velocity phase-space distribution of the particles and allows for an exact expression of the back-reaction from the particles from the action-reaction principle. It is shown that global quantities such as total energy and enstrophy monotonically decrease with the mass ratio, as well as Ekman and molecular dissipation. At large Stokes numbers, the effect of particles is explained in terms of an effective additional Ekman friction. This has the consequence to steepen the power-law behavior of the enstrophy spectrum. The impact on the intermittent properties are also examined. In particular, it is shown that the multiscaling properties of the vorticity structure function follow a simple renormalisation given by second-order statistics. At scales within the dissipative range, this renormalisation stops working. The probability distribution function of the vorticity gradients display tails which broaden when the coupling intensity is increased. Finally, it is also shown that two-way coupling enhances particle clustering at large Stokes numbers. This effect can be attributed to a decrease of the actual particles inertia, making them more likely to be ejected from coherent vortices and to form preferential concentrations.

Key words: turbulence modulation, particle-laden flows, turbulent flows, multiphase flows modelling

1. Introduction

Numerous natural and industrial situations involve particle-laden flows. These include pollutant transport (Gyr & Rys 2013), plankton dynamics and accumulation (Durham *et al.* 2013), the modelling of fluidized beds in pharmaceutical or mining manufacturing processes (Curtis & Van Wachem 2004) and dilute spray combustion (Li *et al.* 2012). Finite-inertia particles display the important feature of detaching from the flow streamlines. Those heavier than the surrounding fluid (such as dust or liquid droplets in gases) tend to be ejected from vortices by centrifugal forces and form preferential concentrations in strain-dominated regions. This effect is expected to enhance turbulent collisions and, as a matter of fact, to trigger coalescences between droplets and accelerate the initiation of rain in warm clouds (Falkovich *et al.* 2002; Lanotte *et al.* 2009). During this small-scale process, particle with very different histories can possibly meet to collide, bringing together different turbulent velocities (Gustavsson *et al.* 2008) or non-trivial inertial-range correlations of transport (Bec *et al.* 2016). Consequently, this microphysical effect is strongly coupled to the full range of length and timescales that are excited by turbulent fluctuations, leading to significant difficulties in establishing reliable quantitative models.

The back reaction exerted by the particles onto the fluid gives rise to an extra complexity in modelling. Such a two-way coupling typically arises when the mass fraction of the dispersed particle phase relative to the carrier fluid is sufficient to provoke momentum exchanges with the surrounding medium (Elghobashi 1994), resulting in a modulation of the turbulent flow. The understanding of the mechanisms at play, even at a purely qualitative level, is still insufficient for developing efficient models of relevance to applications. An instance where turbulence modulation plays a key role is in the process of planet formation during the early stages of the Solar system. It has indeed been shown that the interactions between dust and gas in protoplanetary disks create an instability facilitating grain clumping (Johansen & Youdin 2007; Jacquet *et al.* 2011) and we are still lacking quantitative predictions on how much this enhances the formation of planetesimals. Many questions on the modelling of turbulence modulation by particles remain also open in engineering applications. These include turbulent sprays (Jenny *et al.* 2012) or fuel droplets in combustion chambers (Post & Abraham 2002), where two-way coupling is expected to enhance heat transfers and macroscopic chemical reaction rates.

The development of a comprehensive phenomenology and of efficient models is hindered by the complexity of the problem. A source of complexity comes from the possibility or not to model the particle dynamics itself. For instance, when their sizes are comparable or larger than the smallest active scale of the fluid flow (*e.g.* the Kolmogorov dissipative scale in three-dimensional turbulence), determining particles dynamics requires fully resolving the fluid flow around them and integrating viscous strain and pressure at their surface to obtain net forces. Local modulation of the flow around the particles might then be assessed, as well as modification of global quantities. Various numerical techniques have been developed to the end of studying the effects induced by such finite-size particles. Immersed boundaries (Lucci *et al.* 2010; Cisse *et al.* 2013), two-fluid level-set (Sabelnikov *et al.* 2014; Loisy & Naso 2016) and Lattice-Boltzmann methods have already been used in this context (Poesio *et al.* 2006; Gao *et al.* 2013), allowing one to reach volume fractions of the order of 2-10 % (see, *e.g.*, Ten Cate *et al.* 2004). However, the number of resolved particles is generally limited because of the high computational demand and the effects on turbulence do not manifest in a sufficiently intelligible manner. Experiments have been conducted in order to span higher values of the dispersed phase volume fraction. It was found that in wall flows, the presence of particles can either trigger or prevent the transition to turbulence depending on the size and volume fraction of the particles (Matas *et al.* 2003). In settings closer to homogeneous isotropic turbulence (as, *e.g.*, in a von Karman flow), it was observed that large neutrally buoyant particles attenuate turbulent fluctuations without affecting their nature: The distribution of acceleration statistics and the velocity structure functions are only affected by the modifications of the Reynolds number (Cisse *et al.* 2015). In addition, these measurements suggest a mechanism of particle shedding that prevent turbulent fluctuations from reaching the center of the experiment, advocating that the attenuation of turbulence might strongly depend on the particle size, on its mass, and on the way kinetic energy is injected in the flow. On this account, and as stressed in Saber *et al.* (2015), important efforts are still needed in order to build up a phenomenological outlook and to understand whether or not any universal ideas and concepts can be sketched for the modulation of turbulence by particles.

Such difficulties put forward the need to investigate in more details simpler and more easily controllable situations. A straightforward choice is to focus on very small heavy spherical particles whose dynamics is fully determined by a unique parameter, the Stokes number St. Experimental and numerical studies suggest that the effect of small particles is to attenuate turbulence intensity at low inertia, *i.e.* $St \lesssim 1$ or low volume fractions, and to enhance it when $St \gtrsim 1$ and inertia effects become large (Mandø 2009; Patro & Dash 2014). Furthermore, modulation is not necessarily isotropic nor homogeneous in space. Gualtieri *et al.* (2011) performed simulations with particles with St of order one and that are suspended in a flow with mean shear. The effect of particles exhibits a strong anisotropy: The kinetic energy spectrum displays an excitation of the small scales that clearly indicates that two-way coupling is responsible for a transfer of large-scale anisotropies to the small scales at the expense of the inertial-range energy content. Yang & Shy (2005) investigated intermediate Stokes number ($0 < St < 2$) and computed a wavelet-based energy spectrum. They observed an increase of the turbulent fluctuations at high frequencies, maximal at $St \sim 1$, and more pronounced in the direction of gravity. Hwang & Eaton (2006) considered suspensions with a larger value of the Stokes number ($St \sim 50$) and showed a decrease of the total kinetic energy with increasing mass load. They observed a uniform attenuation across the scales in the horizontal plane and a dissipation less pronounced in the vertical direction.

An advantage of considering small particles is that they are generally associated with a low Reynolds number, so that the forces exerted on them by the fluid can be expressed explicitly. While this could seems as a strong point to perform efficient numerical simulations, it actually leads to other difficulties. In principle, considering small particles foster the use of Euler-Lagrange numerical approaches and thus of a point-force method for the back reaction on the flow. This method originates from the particle-in-cell method (Crowe *et al.* 1977) and consists in considering particles as punctual sources and sinks of momentum. The total forces exerted on the fluid phase by all the particles in a given mesh cell must then be distributed to the neighbouring grid points. As a result, this method is strongly grid-dependent (Balachandar 2009). It greatly relies on the number of particles in each cell, and can suffers from a lack of numerical convergence when this number is too low (Garg *et al.* 2009). This effect can be particularly penalizing in the case of particles with inertia that exhibit preferential concentration and distribute in space in a very non-uniform manner. Progress are however made in order to improve Euler-Lagrange approaches. For instance, Gualtieri *et al.* (2015) used physical arguments to propose a specific regularisation method that is expected to remove most drawbacks linked to the extrapolation of the point-force to the fluid gridpoints. Despite such developments, Euler-Lagrange methods might still not fit situations requiring by essence a large number of particles. These include of course situations where particles are not sufficiently dilute but also cover the case of particles with large Stokes numbers. Such particles are indeed known to develop a broad dispersion in velocity and a grounded representation requires sufficiently many particles to map the full position-velocity phase-space.

The objective of the work reported here is twofold. The first target is to concentrate on a case amenable to a systematic analysis, namely the modulation of two-dimensional incompressible turbulence by small heavy particles. We focus on the direct cascade, where enstrophy (squared vorticity) injected at the large scales, is transfered through the scales, before being dissipated by molecular viscosity (see Boffetta & Ecke 2012, for a review). Besides its relevance to horizontal, large-scale geophysical motions (see, *e.g.*, Danilov & Gurarie 2000), two-dimensional turbulence profits from being easy to simulate numerically. On the one hand, this allows for broadly exploring the dependence upon the two parameters which characterise the coupling with particles, namely their

Stokes number St and their total mass fraction ϕ_{m}. On the other hand, undemanding simulations enable integrations over very long periods, typically of several hundreds of large-eddy turnover times, a necessary duration to guarantee the convergence of large-scale quantities and to gather accurate statistics. The other goal of this work is to propose and exploit a new Eulerian numerical technique that copes with the difficulties encountered when using Lagrangian point-force methods and allows for considering the effect of particles with moderately large Stokes numbers. We make use of the method introduced in Laenen *et al.* (2016), which integrates explicitly the kinetic equation for the particle population without introducing any approximation on the dynamics.

The paper is organised as follows. We formulate the dynamical model in §2 and describe the numerical method in §3. Next, we describe in §4 the effects of two-way coupling on the statistics of the fluid energy and enstrophy. We demonstrate that the presence of large-Stokes-number particles leads to decrease both the dissipation rate and the global values of energy and enstrophy, and concomitantly steepen the corresponding power spectra. Conversely, the feedback of particles with $St \sim 1$ causes an increase of enstrophy dissipation and is responsible for an injection of energy at intermediate scales. In §5, we then present results on small-scale, higher-order statistics. We show that intermittency increases with the particle mass loading. In particular, the probability density functions of the vorticity gradients are found to develop larger tails when the coupling with particles is strengthened. In §6 we report measurements on particle clustering and find that the coupling can either enhance or deplete it depending on the value of the Stokes number. We finally draw some concluding remarks in §7

2. Phase-space description

As stated in the Introduction, we adopt in this work a description of particle suspensions in terms of fields. This method does not rely on approximating the particle dynamics in terms of a particle velocity field but considers the full phase-space distribution $f(\boldsymbol{x},\boldsymbol{w},t)$ of particles positions and velocities. This quantity is defined as

$$f(\boldsymbol{x},\boldsymbol{w},t) = \frac{1}{N_p}\sum_{i=1}^{N_p} \boldsymbol{\delta}(\boldsymbol{X}_i(t)-\boldsymbol{x})\,\boldsymbol{\delta}(\boldsymbol{V}_i(t)-\boldsymbol{w}), \tag{2.1}$$

where N_p is the total number of particles in the domain, $\boldsymbol{X}_i$ and $\boldsymbol{V}_i$ are the position and the velocity, respectively, of the i-th particle. Note that the phase-space distribution f is normalized in such a way that its integral over positions and velocities is equal to unity. The evolution of f is governed by the Liouville equation, which expresses conservation in phase space, namely

$$\partial_t f + \boldsymbol{\nabla}_x \cdot (\boldsymbol{w}\, f) + \boldsymbol{\nabla}_w \cdot (\boldsymbol{F}_{\mathrm{f}\to\mathrm{p}}\, f) = 0 \tag{2.2}$$

where $\boldsymbol{F}_{\mathrm{f}\to\mathrm{p}}(\boldsymbol{x},\boldsymbol{w},t)$ denotes the sum of the forces acting at time t on a particle at position $\boldsymbol{x}$ with velocity $\boldsymbol{w}$. These various forces can be very complex when including all possible effects, such as buoyancy, Basset–Boussinesq history forces due to the interaction of the particle with its own wake, the added-mass factor, Faxén corrections, etc. (see Maxey & Riley 1983, for details). We focus here on the case of small, very heavy, low-Reynolds-number spherical particles. Namely, we assume that the particles radius a is much smaller than any active scale of the flow and that their material mass density ρ_{p} is much larger than the fluid density ρ_{f} and or the sake of simplicity we also neglect the effect of gravity.

Under these assumptions, the forces reduce to the Stokes drag:

$$\boldsymbol{F}_{\mathrm{f}\to\mathrm{p}}(\boldsymbol{x},\boldsymbol{w},t) = -\frac{1}{\tau_{\mathrm{p}}}\left(\boldsymbol{w}-\boldsymbol{u}(\boldsymbol{x},t)\right), \tag{2.3}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ denotes the fluid velocity evaluated at the particle position and τ_p is the particle response time given by $\tau_{\mathrm{p}} = 2a^2\rho_{\mathrm{p}}/(9\,\rho_{\mathrm{f}}\,\nu)$, with ν the fluid kinematic viscosity.

The fluid velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ evolves according to the two-dimensional incompressible Navier-Stokes equations

$$\partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla_x)\,\boldsymbol{u} = -\nabla_x p - \alpha\,\boldsymbol{u} + \nu\nabla_x^2\boldsymbol{u} + \boldsymbol{F}_u + \boldsymbol{F}_{\mathrm{p}\to\mathrm{f}}, \quad \nabla_x\cdot\boldsymbol{u} = 0. \tag{2.4}$$

The flow is sustained in a stationary turbulent regime by the external Gaussian force $\boldsymbol{F}_u$ which is assumed to be concentrated on the large spatial scales and white noise in time. The linear damping $-\alpha\,\boldsymbol{u}$ represents Ekman friction and prevents the pile-up of kinetic energy at large scales. The force $\boldsymbol{F}_{\mathrm{p}\to\mathrm{f}}$ exerted by the particles onto the fluid follows from the action-reaction principle and reads

$$\boldsymbol{F}_{\mathrm{p}\to\mathrm{f}}(\boldsymbol{x},t) = -\phi_{\mathrm{m}}\int \boldsymbol{F}_p(\boldsymbol{x},\boldsymbol{w},t)\,f(\boldsymbol{x},\boldsymbol{w},t)\;\mathrm{d}^2w \tag{2.5}$$

This volume forcing depends on the mass ratio parameter $\phi_{\mathrm{m}} = N_p\,\mathcal{V}_{\mathrm{p}}\,\rho_p/(\mathcal{V}\,\rho_f)$, where $\mathcal{V}_{\mathrm{p}}$ is the volume of a single particle, N_p is the total number of particles in the domain whose volume is denoted $\mathcal{V}_f$. As we consider very heavy particles ($\rho_p/\rho_f \gg 1$), one can possibly get finite values of ϕ_{m} even if the volume fraction $N_p\mathcal{V}_{\mathrm{p}}/\mathcal{V}$ is small.

We further introduce the particle density n_{p} and the average particle velocity $\boldsymbol{v}_{\mathrm{p}}$ as

$$n_{\mathrm{p}}(\boldsymbol{x},t) = \int f(\boldsymbol{x},\boldsymbol{w},t)\;\mathrm{d}^2w, \tag{2.6}$$

$$n_{\mathrm{p}}(\boldsymbol{x},t)\,\boldsymbol{v}_p(\boldsymbol{x},t) = \int \boldsymbol{w}\,f(\boldsymbol{x},\boldsymbol{w},t)\;\mathrm{d}^2w \tag{2.7}$$

Note that n_{p} is not a mass density field. Because of the definition of f, it actually counts the number of particles (and is rather a numer density) but it is normalised as $\int n_p(\boldsymbol{x},t)\;\mathrm{d}^2x = 1$. It is worth keeping in mind that for particles with large Stokes numbers, the actual velocities at a given spatial position can be very dispersed and generally differ from the average velocity field $\boldsymbol{v}_p$.

Using (2.6), (2.7), and the Stokes drag expression (2.3), one obtains the following expression for the force exerted by the particles onto the fluid

$$\boldsymbol{F}_{\mathrm{p}\to\mathrm{f}}(\boldsymbol{x},t) = -\frac{\phi_{\mathrm{m}}}{\tau_{\mathrm{p}}}\,n_p(\boldsymbol{x},t)\;\left[\boldsymbol{u}(\boldsymbol{x},t)-\boldsymbol{v}_{\mathrm{p}}(\boldsymbol{x},t)\right]. \tag{2.8}$$

Usually, the dynamics of incompressible two-dimensional flows is formulated in terms of the streamfunction $\psi(\boldsymbol{x},t)$, such that $\boldsymbol{u} = (\partial_2\psi, -\partial_1\psi)$, the indices 1, 2 being associated to the two spatial dimensions, and of the scalar vorticity $\omega = \nabla_x\times\boldsymbol{u} = \partial_1u_2 - \partial_2u_1 = -\nabla_x^2\psi$. The vorticity-streamfunction formulation is obtained from the curl of Navier–Stokes equation ((2.4)) and reads

$$\partial_t\omega + \mathcal{J}(\psi,\omega) = \nu\nabla_x^2\omega - \alpha\,\omega + F_\omega - \frac{\phi_{\mathrm{m}}}{\tau_{\mathrm{p}}}\;\left[n_{\mathrm{p}}\,\nabla_x\times(\boldsymbol{u}-\boldsymbol{v}_{\mathrm{p}}) + (\boldsymbol{u}-\boldsymbol{v}_{\mathrm{p}})\cdot\nabla_x n_{\mathrm{p}}\right], \tag{2.9}$$

where we introduced the two-dimensional Jacobian $\mathcal{J}(\psi,\omega) = \partial_1\psi\,\partial_2\omega - \partial_2\psi\,\partial_1\omega$ and $F_\omega = \nabla_x\times\boldsymbol{F}_u$.

In the sequel we investigate the modification by particles of a fundamental turbulent state. This primary regime is obtained by setting $\phi_{\mathrm{m}} = 0$ and fixing the characteristics of

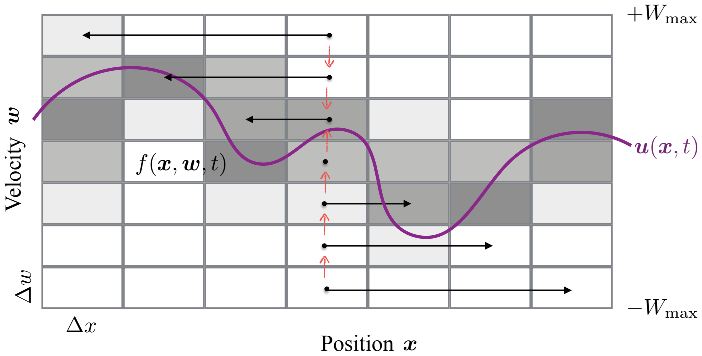


FIGURE 1. Sketch of the algorithm used to update the position-velocity phase-space particle distribution $f(\boldsymbol{x},\boldsymbol{w},t)$. The positions are discretized with a resolution Δx and the velocities, bounded between $-W_{\max}$ and $+W_{\max}$ with a resolution Δw. At each time step,the dynamics is split in two operations: advection with the corresponding velocity (horizontal arrows) and relaxation to the fluid velocity (vertical arrows).

the external forcing $\boldsymbol{F}_u$ and the values of the dissipative constants ν and α. It is chosen to correspond to a developed direct enstrophy cascade from which a root-mean-squared vorticity $\omega_{\rm rms}$ can be measured. This value, obtained with no coupling to the particle phase, serves as a reference. Deviations due to the coupling between the particle and the fluid phases are obtained by varying only two control parameters: the non-dimensional mass loading $\phi_{\rm m}$ defined above and the Stokes number $St_0 = \tau_{\rm p}\,\omega_{\rm rms}$ obtained by non-dimensionalising the particle response time by the root-mean-squared value of the vorticity obtained with no coupling.

3. Numerical method

The incompressible fluid flow is integrated numerically using a spectral, fully de-aliased solver based on the vorticity-streamfunction formulation (2.9) of the Navier-Stokes equations. Time-marching is done using a second-order Runge–Kutta scheme. This equation is coupled to the time evolution of the particle phase-space distribution f given by the Liouville equation (2.2). This integration is performed by means of the *lattice-particle method* introduced by Laenen *et al.* (2016) and which has been shown to reproduce well the concentration and velocity properties of inertial particles in random and turbulent flows.

The lattice-particle algorithm is inspired from the lattice-Boltzmann method (see, *e.g.*, Succi 2001, for a review). The idea is to approximate the distribution f as a piecewise constant field discretized in phase-space on a square lattice with resolution $N_x^2 \times N_w^2$. For a square periodic spatial domain of size L, the resolution in particle positions is $\Delta x = L/N_x$. The velocity domain is the bounded square $[-W_{\max}, +W_{\max}]^2$ and is divided in squares of size $\Delta w = 2W_{\max}/N_w$. The time step size is chosen such that $\Delta x = \Delta w\,\Delta t$. The algorithm to update f between two consecutive time steps consists in two successive operations (illustrated in Fig. 1 in the one-dimensional case for simplicity). The first step consists in advecting on the spatial domain the mass situated in the cell $\boldsymbol{x} = (i_1\,\Delta x, i_2\,\Delta x)$ with velocity $\boldsymbol{w} = (j_1\Delta w, j_2\Delta w)$. This mass is displaced to the cell $\boldsymbol{x}' = ((i_1 + j_1)\,\Delta x, (i_2 + j_2)\,\Delta x)$ associated to the same velocity $\boldsymbol{w}$. The second step consists in updating the value of f according to the conservation law (2.2) to account for the relaxation of the particle velocity to that of the fluid at the same location. This is performed using a finite-volume scheme in the velocity direction, which is based on

$\phi_{\rm m}$	$\tau_{\rm p}$	E	Z	η_C	η_ν	η_α
0	0.5	0.97	2.1	0	0.047	0.021
0.1	0.5	0.43	1.6	0.033	0.019	0.016
0.2	0.5	0.34	1.4	0.038	0.015	0.014
0.3	0.5	0.29	1.3	0.039	0.016	0.013
0.4	0.5	0.22	1.2	0.038	0.018	0.012
0.5	0.5	0.19	1.1	0.036	0.019	0.011
0	2	1.10	2.4	0	0.047	0.024
0.1	2	0.40	1.8	0.033	0.021	0.018
0.2	2	0.26	1.1	0.046	0.011	0.011
0.3	2	0.27	1.2	0.05	0.0083	0.012
0.4	2	0.19	0.88	0.054	0.0055	0.0088
0.5	2	0.16	0.83	0.057	0.0049	0.0083
0	8	1.10	2.3	0	0.047	0.023
0.1	8	0.38	1.3	0.021	0.032	0.013
0.2	8	0.30	1.2	0.033	0.020	0.012
0.3	8	0.23	1.0	0.044	0.016	0.010
0.4	8	0.19	0.89	0.047	0.0097	0.0089
0.5	8	0.17	0.83	0.051	0.0072	0.0083

TABLE 1. Parameters used for the numerical simulations. $\phi_{\rm m}$ is the particle mass fraction, $\tau_{\rm p}$ their response time, E is the measured average kinetic energy of the fluid and Z its average enstrophy. η_C is the average enstrophy dissipation due to the coupling, η_ν the average enstrophy dissipation due to viscosity and η_α the average enstrophy dissipation due to Ekman friction. In all cases, the spatial resolution, *i.e.* the number of spatial collocation points for the fluid phase and the number of spatial cells for the solid phase, is always equal to $N_x^2 = 1024^2$. The number of velocity cells for the solid phase is fixed to $N_w^2 = 9^2$. The kinematic viscosity of the fluid is fixed to $\nu = 5 \times 10^{-5}$ and the Ekman friction parameter to $\alpha = 0.005$.

estimating mass fluxes between adjacent cells. We make use of a positivity-preserving flux-limiter algorithm in order to avoid diverging gradients in the distribution (see, *e.g.*, LeVeque 2002).

One of the key points of this numerical method is to provide *a priori* a correct guess for the maximum velocity of the particles $W_{\rm max}$. To avoid arbitrary cautious choices that might lead to non-optimal resolutions, we adopt an adaptive re-meshing of the velocity grid, by keeping fixed N_w and imposing $W_{\rm max}$ to be at least twice the root-mean squared value of the particle velocity computed from the distribution f. The time step Δt is then adapted to maintain the constraint $\Delta x = \Delta w\, \Delta t$.

Here and in the following, the system is discretized on a square domain of size $L = 2\pi$ with periodic boundary conditions. The particles are integrated with a resolution $N_x^2 \times N_v^2 = 1024^2 \times 9^2$ and the fluid velocity discrete Fourier transform is computed with 512^2 wavenumbers (corresponding to an effective resolution of 1024^2 collocation points). The flow is sustained by a large-scale Gaussian random forcing, acting on modes $\boldsymbol{k}$ with moduli satisfying $1 < |\boldsymbol{k}| < 4$. The simulations are initialized with the fluid flow at rest and uniformly distributed particles with zero velocity. Once the flow and the particles have reached a statistically stationary regime, the statistics are performed over 300 large-eddy turnover times $T = L/u_{\rm rms}$. The parameters of the simulations are reported in Table 1.

In Figure 2 we compare typical snapshots (taken at random times in the stationary regime) of the fluid vorticity $\omega(\boldsymbol{x}, t)$ (upper panels), along with the particle density field $n_{\rm p}(\boldsymbol{x}, t)$ integrated over velocities (lower panels), with and without back-reaction

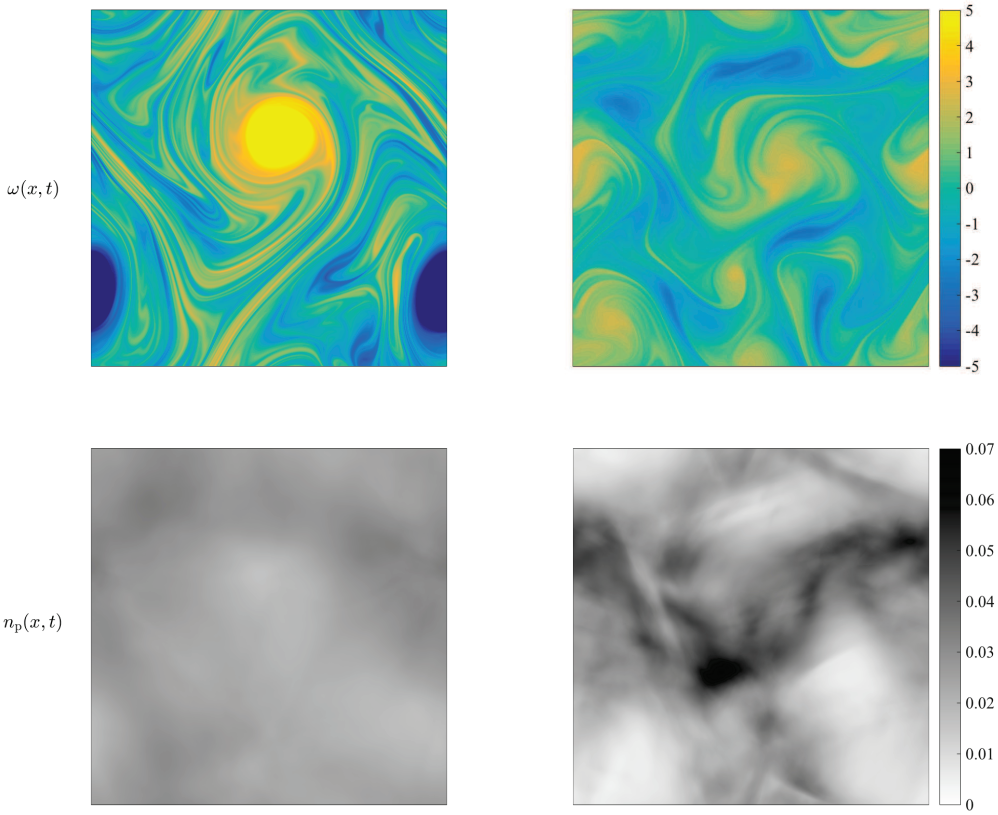


FIGURE 2. Snapshots of the fluid vorticity (upper panels) and of the particle density (lower panel) illustrating the effect of the coupling for $St_0 \simeq 17$. The left panels correspond to the case with no coupling between the two phases ($\phi_{\rm m} = 0$). The right panels are for the case with a moderately strong coupling ($\phi_{\rm m} = 0.6$). The color scales are the same for the left and right figures.

from the particles and in both case for $St_0 \simeq 17$. In the case without coupling (left panels), the fluid flow is dominated by large eddies and particles are ejected from them to concentrate in the high-strain filamentary regions outside. This is a clear illustration of preferential concentration by centrifugal effects. Note that for such a high value of the Stokes number, the particle density is nearly uniform. Preferential concentration is indeed weak as the particles tend to follow ballistic trajectories, ignoring the vortices. As soon as the coupling is turned on (right panels), one clearly detect a significant damping of the large scales, still keeping larger densities outside the eddies. This qualitative observation can be quantitatively confirmed by measuring the energy-containing scale defined as $L_E(\phi_{\rm m}) = (\sum_k E(k)/k)/(\sum_k E(k))$ where $E(k)$ is the energy power spectrum of the fluid velocity $\boldsymbol{u}$. This lengthscale, which is not shown here, is a decreasing function of the mass load $\phi_{\rm m}$. As can been seen from the lower panels of Fig. 2, the particle density pictures an enhancement of the fluctuations, indicating an increase of clustering due to the coupling.

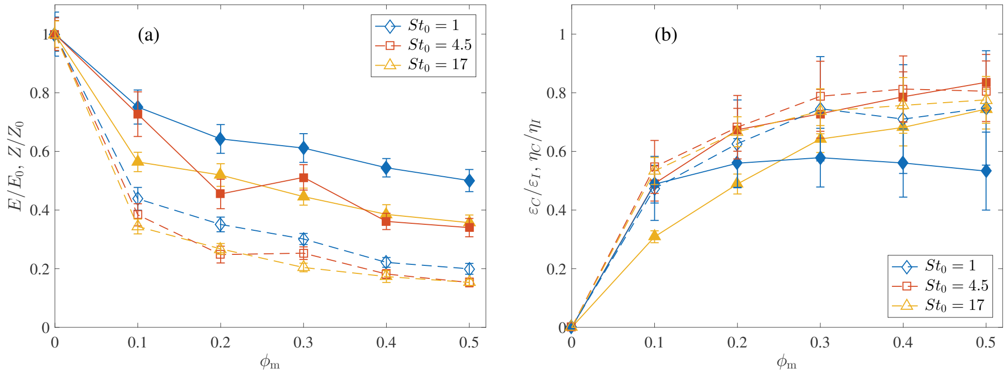


FIGURE 3. (a) Total energy E (dashed lines, open symbols) and enstrophy Z (solid lines, filled symbols) normalized by their respective values E_0 and Z_0 obtained with no coupling, and represented as a function of the mass loading $\phi_{\rm m}$ for various values of the Stokes number as labeled. (b) Average rates of energy exchange ε_C (dashed lines, open symbols) and of enstrophy exchange η_c (solid lines, filled symbols) between the fluid and the particle phases. These quantities, normalized to their respective injection rates ε_I and η_I, are represented as a function of the mass loading $\phi_{\rm m}$ for the different values of the Stokes number.

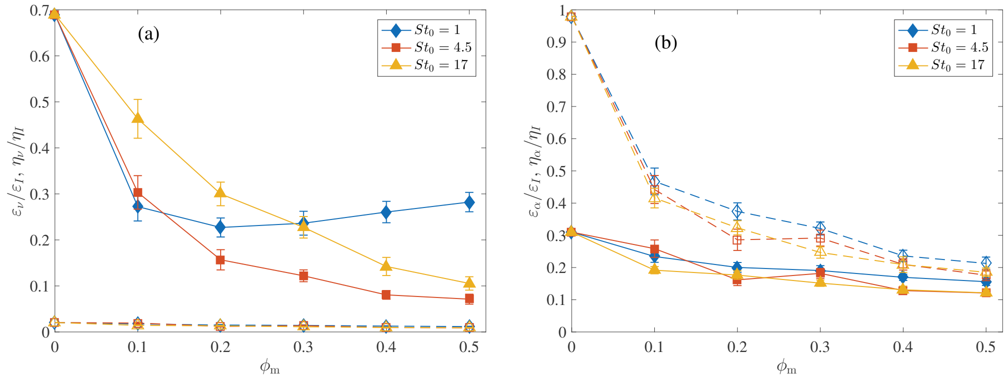


FIGURE 4. (a) Average dissipation rates of energy ε_ν (dashed lines, open symbols) and of enstrophy η_ν (solid lines, filled symbols) due to molecular viscosity, normalized to the injection rates ε_I and η_I, and represented as a function of the mass loading $\phi_{\rm m}$ for the different values of the Stokes number. (b) Same for the dissipation rates ε_α and of η_α due to the linear Ekman friction term.

4. Modification of energy and enstrophy budgets

We report in this section measurements on how the energy and enstrophy of the fluid are affected by the coupling with the particle phase. The feedback of the particles has strong consequences on the global kinetic energy $E = (1/2)\langle|\boldsymbol{u}|^2\rangle$ and the enstrophy $Z = (1/2)\langle\omega^2\rangle$, where the brackets $\langle\cdot\rangle$ denotes averages over space and time. In particular, as seen in Fig. 3(a), we find that both quantities decrease at increasing the mass loading ratio $\phi_{\rm m}$.

The reduction of the kinetic energy of the fluid phase can be understood by considering the global energy balance, which in the statistically steady state reads

$$\varepsilon_I = \varepsilon_\alpha + \varepsilon_\nu + \varepsilon_C. \tag{4.1}$$

Here, $\varepsilon_I = \langle \boldsymbol{F}_u \cdot \boldsymbol{u}\rangle$ denotes the average energy input, $\varepsilon_\alpha = \alpha\,\langle|\boldsymbol{u}|^2\rangle$ is the energy dissipation rate due to friction, $\varepsilon_\nu = \nu \langle |\nabla \boldsymbol{u}|^2\rangle$ is the energy dissipation rate due to viscosity, and $\varepsilon_C = \langle \boldsymbol{F}_{\mathrm{p}\to\mathrm{f}} \cdot \boldsymbol{u}\rangle = -\phi_{\mathrm{m}} \langle n_{\mathrm{p}}(|\boldsymbol{u}|^2 - \boldsymbol{u}\cdot\boldsymbol{v}_{\mathrm{p}})\rangle/\tau_{\mathrm{p}}$ is the exchange rate of kinetic energy from the fluid to the particles. One can show analytically (see Appendix for details) that the coupling between the fluid and particle acts on average as a dissipative term for the kinetic energy of the flow, that is $\langle \boldsymbol{F}_{\mathrm{p}\to\mathrm{f}} \cdot \boldsymbol{u}\rangle > 0$. This can be understood intuitively from the simple consideration that the particles move because the flow gives them a fraction of its kinetic energy. This is confirmed by the numerical simulations, which shows a significant transfer of energy from the fluid to the particle as soon as $\phi_{\mathrm{m}} > 0$. This effect increases with the mass loading ϕ_{m}, as can be seen in Fig. 3(b). Already when the mass ratio is $\phi_{\mathrm{m}} = 0.1$, we find that more than 50% of the kinetic energy injected in the system is transferred to the particles. The viscous dissipation rate ε_ν, being proportional to the enstrophy Z, decreases with ϕ_{m} as confirmed in Fig. 4(a). As usual in the two-dimensional direct enstrophy cascade, its contribution to the energy balance is sub-dominant and this effect is unvarying when the coupling increases: We always observe $\varepsilon_\nu/\varepsilon_I < 2\%$ with almost no dependence on ϕ_{m} or St_0. The dissipation due to friction ε_α is actually dominating the energy budget. Because it is proportional to the kinetic energy, we expect it to decrease with ϕ_{m}, as confirmed in Fig. 4(b).

In analogy to the quantities introduced in the energy balance (4.1), we define the enstrophy input rate $\eta_I = \langle F_\omega\, \omega\rangle$, the enstrophy dissipation rate due to viscosity, $\eta_\nu = \nu \langle |\boldsymbol{\nabla}\omega|^2\rangle$, the enstrophy dissipation rate due to friction $\eta_\alpha = \alpha \langle \omega^2\rangle$, and the exchange rate of enstrophy with the particle phase $\eta_C = \langle \nabla \times \boldsymbol{F}_{\mathrm{p}\to\mathrm{u}}\omega\rangle$. Their values obtained from the various simulations are reported in Tab. 1. As for energy, these quantities enter in the enstrophy budget, which reads in the statistically steady

$$\eta_I = \eta_\alpha + \eta_\nu + \eta_C \tag{4.2}$$

The dissipation rate of enstrophy due to friction η_α is proportional to the enstrophy itself, and therefore displays the same slowly decreasing behavior as a function of ϕ_{m}, as shown in Fig. 4(b).

The contributions η_ν and η_C stemming from molecular viscosity and coupling with the particle phase reveal a much richer phenomenology. We see from Fig. 4(b) that the viscous dissipation decreases with ϕ_{m} at large values St_0, while it displays a non-monotonic behavior in the case $St_0 \simeq 1$. Similarly, the transfer from the enstrophy of the fluid to the particles increases with ϕ_{m} only at large St_0, while the case $St_0 \simeq 1$ displays a non-monotonic behavior, as visible in Fig. 3(b). As we will discuss in the following, the increase at large ϕ_{m} of both the enstrophy viscous dissipation comes from a strong coupling with the particle phase that enhances the enstrophy transfers to the small scales.

In our simulations the flow is sustained by a large-scale forcing, which gives rise to an enstrophy cascade toward viscous scales. We investigate here in more details how the coupling with the particle phase influences this process by measuring the scale-by-scale enstrophy budget. The equation for the enstrophy spectrum $Z(k)$ reads:

$$\partial_t Z(k) = -T(k) - 2\alpha\, Z(k) + \mathit{\Phi}_Z(k) - 2\nu\, k^2\, Z(k) + C(k), \tag{4.3}$$

where $\mathit{\Phi}_Z(k) = \langle \mathcal{F}[F_\omega\, \omega]\rangle$ is the spectral injection of enstrophy by the forcing, $\mathcal{F}[\cdot]$ denoting the Fourier transform. This contribution is concentrated at scales $k < 4$. The term $C(k) = \langle \mathcal{F}[\nabla \times \boldsymbol{F}_{\mathrm{p}\to\mathrm{u}}\, \omega]\rangle$ is the spectral contribution due to the coupling with particles and $T(k) = \langle \mathcal{F}[\omega\, (\boldsymbol{u}\cdot\boldsymbol{\nabla}_x \omega)]\rangle$ is the spectral enstrophy transfer due to the non-linear advection term. The averages are taken over time and the wave-number shell of modulus $|\boldsymbol{k}| = k$. In the statistically stationary regime, $\partial_t Z(k) = 0$ and the various terms on the right-hand side compensate each other. In the inertial range and

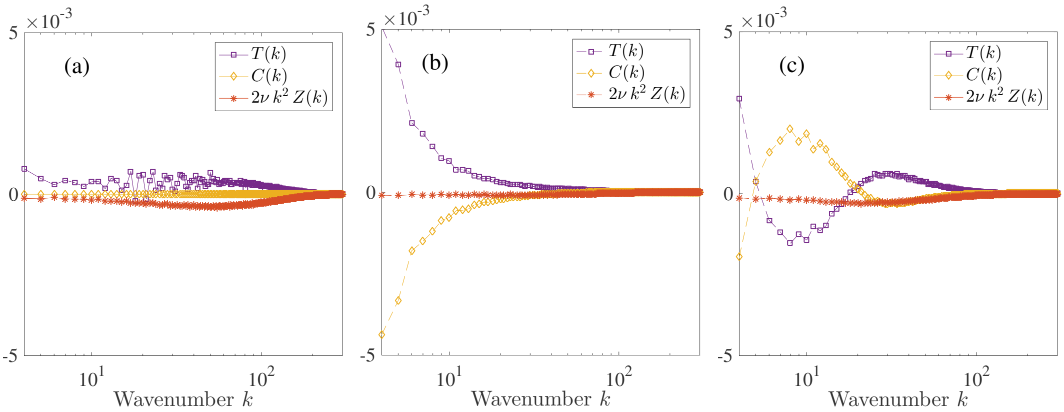


FIGURE 5. Spectral contributions of non-linear transfer $T(k)$, coupling with the particle phase $C(k)$, and viscous dissipation $2\nu k^2 Z(k)$ entering in the enstrophy balance (4.3) for (a) no coupling: $\phi_{\mathrm{m}} = 0$, (b) coupling with a large Stokes number: $\phi_{\mathrm{m}} = 0.5$ with $St_0 \simeq 17$, and (c) coupling with a moderate Stokes number: $\phi_{\mathrm{m}} = 0.5$ with $St_0 \simeq 1$.

in absence of coupling, the non-linear transfer is balanced by viscous dissipation, *i.e.* $T(k) \simeq -2\nu\, k^2\, Z(k)$.

The spectral behaviors of the viscous term, the transfer term and the coupling term are shown in Fig. 5. In the absence of coupling ($\phi_{\mathrm{m}} = 0$, left-most panel), we observe a balance in the inertial and dissipative ranges between the non-linear transfer and the viscous dissipation. In the case of strong coupling and a large value of the Stokes number ($\phi_{\mathrm{m}} = 0.5$, $St_0 \simeq 17$, middle panel), we observe at low wavenumbers an important dissipative (*i.e.* negative) contribution from the term due to coupling, which is balanced by a symmetric positive enhancement in the enstrophy transfer term. It seems in that case that all the effects of the particles back-reaction are present over all scales, even if they monotonically decrease with k. For the case of a smaller Stokes number ($St \simeq 1$, right-most panel), the spectral behavior is much more complex. In this case, the effects of coupling is still dissipative at very low wavenumbers, but becomes positive in an intermediate range of wavenumbers. This means that the interaction between the two phases is responsible for injecting enstrophy at intermediate scales. Consequently, the transfer term also changes sign. In the limit of small Stokes numbers it is possible to show that this phenomenon is connected to the occurrence of small-scales instabilities. The numerical method used in this work is however not satisfactorily addressing the asymptotics of weak particle inertia, so that undertaking this issue requires appropriate developments. This is thus the subject of a separate work that will be published elsewhere.

As observed in Fig. 5(b), the back-reaction of large-Stokes-number particles strongly modifies the enstrophy scale-by-scale budget. The coupling acts over all scales and breaks up the balance between non-linear transfer and viscous dissipation. This effect causes significant changes in the enstrophy power spectrum. Figure 6(a) shows the various enstrophy spectra obtained when varying the mass load ϕ_{m} while keeping the Stokes number constant at a fixed large value $St_0 \simeq 17$. For a better visibility, the spectra are compensated with the slope measured in the uncoupled case, namely $Z(k) \propto k^{-1.4}$. The deviation to the k^{-1} prediction of Kraichnan (1967) is due to the presence of linear damping. It is indeed known that in two-dimensional turbulence with Ekman friction, the competition between the exponential separation of Lagrangian trajectories and the exponential damping of fluctuations is responsible for a strong intermittency, and in particular drastically affects the spectral scaling of enstrophy (Boffetta *et al.* 2005). The

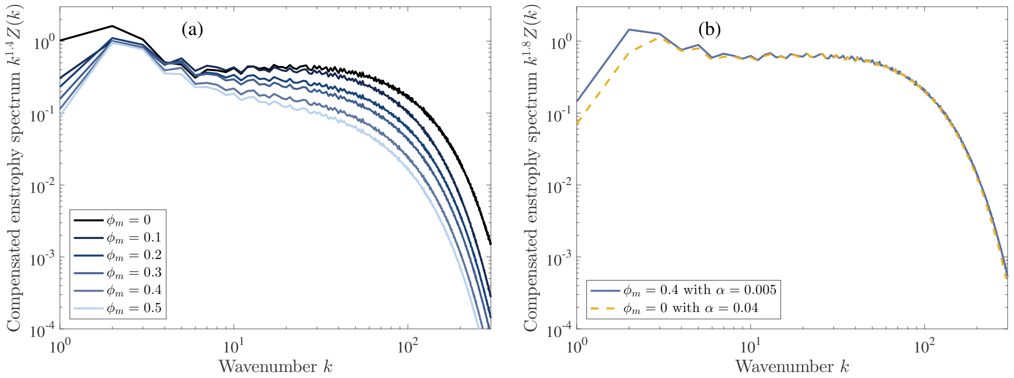


FIGURE 6. (a) Fluid enstrophy spectra for $St_0 \sim 17$ fixed and various mass loads; the spectra are compensated with the power-law $k^{1.4}$ corresponding to the case with no coupling.(b) Comparison of the enstrophy spectra obtained in the coupled case with $St = 17$ and $\phi_{\rm m} = 0.4$ and in the uncoupled case with the effective friction given by (4.4). The spectra have been compensated by $k^{1.8}$ for better visualization.

power-law scaling of $Z(k)$ is steeper than the k^{-1} prediction, *i.e.* $Z(k) \propto k^{-1-\delta}$, and the deviation $\delta > 0$ increases linearly with the Ekman friction coefficient α (Verma 2012).

We see from Fig. 6(a) that when increasing the mass load $\phi_{\rm m}$ and thus the strength of coupling with the particles, an additional steepening of the spectrum power-law exponent occurs. This effect can be explained by an heuristic argument. At very large Stokes, the particles are almost uniformly distributed, $n_p(\boldsymbol{x}, t) \simeq 1/\mathcal{V}$, and their velocity is on average much smaller than that of the fluid $|\boldsymbol{v}_{\rm p}| \ll |\boldsymbol{u}|$. This suggests that the coupling force (2.8) entering in Navier–Stokes equation (2.4) can be approximated as

$$\boldsymbol{F}_{\rm p\to f} = -\frac{\phi_{\rm m}}{\tau_{\rm p}}\, n_p\, [\boldsymbol{u} - \boldsymbol{v}_{\rm p}] \approx -\frac{\phi_{\rm m}}{\tau_{\rm p}\, \mathcal{V}}\, \boldsymbol{u}. \tag{4.4}$$

Coupling thus acts to leading order as an effective linear friction with coefficient $\alpha_{\rm eff} = \phi_{\rm m}/(\tau_{\rm p}\, \mathcal{V})$. From a phenomenological viewpoint, this amounts to say that the particles are seen by the fluid almost like fixed obstacles which are increasing the friction drag.

In order to check the above arguments predicting that the modification of the enstrophy spectrum by particles can be mimicked by an increase of the friction coefficient α, we compare in Fig. 6(b) the results form a simulation with $St = 17$ and $\phi_{\rm m} = 0.4$ with those obtained in the absence of coupling but with the effective friction given by (4.4). The collapse of the two spectra in the scaling range and viscous range is remarkable. This indicates that the effect of the particles on the vorticity filament in the straining regions can be effectively modelled by an increased friction. However we observe some discrepancies at large scales. The increase of the effective friction indeed causes a strong depletion of the large-scale vortices. Conversely, the particles are ejected from the large-scale vortices. Therefore they causes weaker modifications of the flow at those scales.

5. Amplification of intermittency

In previous section, we have focused on the second-order statistics of the velocity and vorticity fields. Here we investigate the effect of two-way coupling on higher-order statistics. In particular, we study the statistics of the vorticity increments $\delta_r\omega = \omega(\boldsymbol{x} + \boldsymbol{r}) - \omega(\boldsymbol{x})$ and of their moments, which define the vorticity structure functions $S_p(r) = \langle |\delta_r\omega|^p \rangle$, where the angular brackets comprise averages over space, time and, by isotropy,

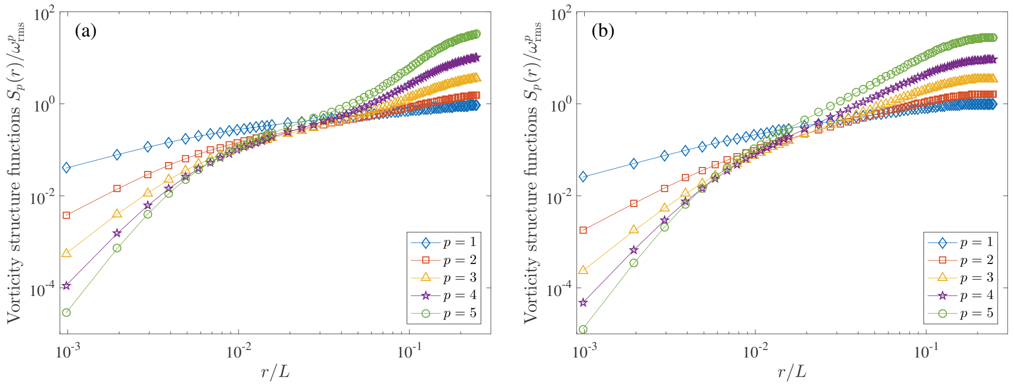


FIGURE 7. Vorticity structure functions of order $p = 1\dots 5$ (as labeled), represented as a function of the separation r in the cases (a) without coupling ($\phi_{\rm m} = 0$) and (b) with $\phi_{\rm m} = 0.5$ and $St_0 \simeq 17$.

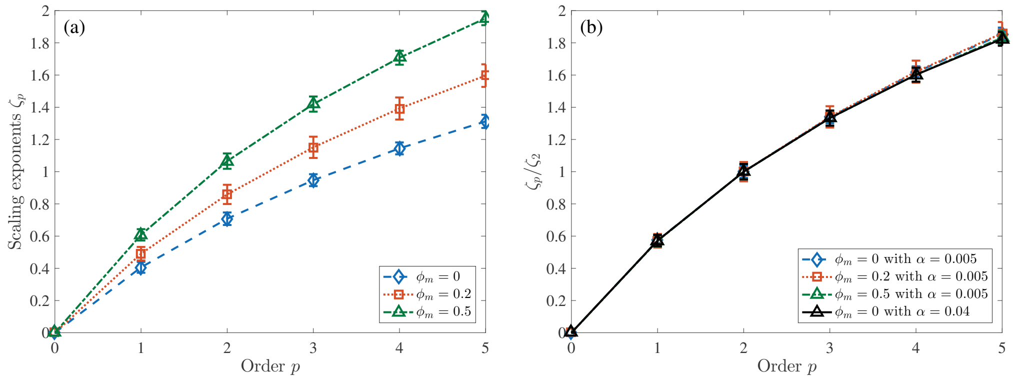


FIGURE 8. (a) Scaling exponents ζ_p of the p-th order vorticity structure function for various values of the mass loading $\phi_{\rm m}$ and for $St_0 \simeq 17$, obtained by averaging the logarithmic derivative $\mathrm{d}\log S_p(r)/\mathrm{d}\log r$ over the range $0.015 < r/L < 0.035$. (b) Same exponents, but this time represented once rescaled with ζ_2 and put together with the exponents obtained in the absence of coupling but with a different value of the friction parameter α.

the angle of the separation vector $\boldsymbol{r}$. We have seen in previous section that at large Stokes numbers, the effect of coupling can be described by a renormalization of the friction coefficient. In the direct enstrophy cascade of two-dimensional turbulence with non-zero large scale friction, vorticity fluctuations have been shown to be intermittent at small-scale. This was conjectured by Bernard (2000) and Nam *et al.* (2000) and studied numerically by Boffetta *et al.* (2002). This implies that their probability distribution differ from a Gaussian and that the exponents ζ_p of the vorticity structure functions, defined by $S_p(r) \propto r^{\zeta_p}$, do not scale linearly with their order p. Such multiscaling properties are however not present for the velocity structure functions (Perlekar & Pandit 2009).

Figure 7 gives a comparison of the vorticity structure functions without and with coupling to a particle phase with $St_0 \simeq 17$. We observe a narrow scaling region $0.015 < r/L < 0.035$. From the analysis of the local slopes in the scaling range, we compute the exponents which are shown in Fig. 8(a). The exponents do not scale linearly with the order p for all values of $\phi_{\rm m}$, indicating the presence of intermittency. In addition, the exponents become larger when increasing the coupling with the particle phase,

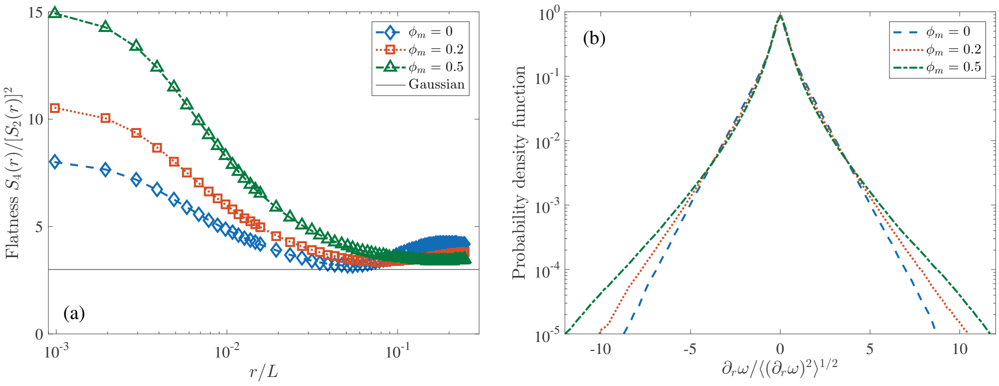


FIGURE 9. (a) Flatness $\mathcal{F}(r) = S_4(r)/[S_2(r)]^2 = \langle\delta_r\omega^4\rangle/\langle\delta_r\omega^2\rangle^2$ of the distribution of vorticity increments as a function of the separation r and for $St_0 \simeq 17$ and various values of the mass load. (b) Probability density function of the vortcity gradients normalized to unit variance represented here for $St_0 \simeq 17$ and various values of the mass load.

in agreement with what was observed in previous section for second-order statistics. However, the dependence on ϕ_m remarkably disappears once the exponents are rescaled by the value of ζ_2, a representation which is somewhat equivalent to the extended self-similarity approach. As can indeed be seen in Fig. 8(b), the various measurements of ζ_p/ζ_2 collapse, within statistical uncertainty, onto a unique universal curve. The measurement obtained from a different value of the linear damping coefficient α are put on the top of the measurements of Fig. 8(b). They also follow the same universal law. This observation suggests, on the one hand, that all intermittency effects due to Ekman friction on the inertial-range scaling exponents of vorticity are entailed in second-order statistics, *i.e.* one has $\zeta_p(\alpha)/\zeta_2(\alpha) = f(p)$ where $f(p)$ is a universal function independent of the coefficient α and, on the other hand, the coupling with a particle phase associated to a large Stokes number is reproduced at all orders by the effective linear friction expressed in (4.4).

This specific behavior in inertial-range multiscaling properties is not detectable from usual measurements of intermittency. For instance, it is frequent to make use of the flatness of the distribution of increments defined as $\mathcal{F}(r) = S_4(r)/[S_2(r)]^2$, whose discrepancy to the value $\mathcal{F}(r) = 3$ quantifies the deviations from a normal distribution. The changes in behavior of $\mathcal{F}(r)$ at varying the coupling with the particle phase are represented in Fig. 9(a). Above results suggest that, in the inertial range, $\mathcal{F}(r) \sim r^\gamma$ with $\gamma = (f(4) - 1)\,\zeta_2$, with all effects of coupling entailed in the variations of ζ_2. This completely explains the increase of $\mathcal{F}(r)$ observed when fixing r in the inertial range and increasing ϕ_m. However, Figure 9(a) suggests that the effects of the coupling are not limited to the scaling range. At large scales, we observe that the flatness of vorticity increments decreases with ϕ_m. This behavior is connected to the suppression of the large scales eddies, already observed qualitatively in Fig. 2. Conversely, at dissipative scales, we observe that $\mathcal{F}(r)$ attains in all cases a plateau with stronger departures from a Gaussian distribution when the coupling with particles and thus ϕ_m increases. This is confirmed in the limit $r \to 0$ by measuring the probability density functions (pdf) of the vorticity gradient. The results are shown in Fig. 9(b). When ϕ_m increases, the pdfs develop broader tails even if the vorticity gradient is normalized with its standard deviation. This effect suggest that, at difference with the inertial range, the effect of particles on small-scale statistics is not fully determined by second-order quantities.

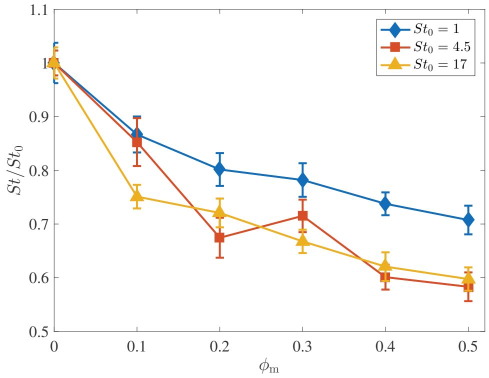


FIGURE 10. Stokes number $St = \tau_\mathrm{p}\,\omega_\mathrm{rms}$ represented here normalized by its reference value St_0 obtained from the uncoupled regime and as a function of the particle mass loading ϕ_m.

6. Enhancement of particle clustering

To go beyond the modulation of turbulence by particles, we address in this section the question of how the particle spatial distribution is itself affected by two-way coupling. A first prediction can be made by recalling that we observed a decrease of the total enstrophy when the particle mass loading is increased. As a result, the time scale of the flow increases and the effective Stokes number experienced by the particles becomes smaller than St_0. This is shown in Fig. 10 which represents the evolution of the actual Stokes number St as a function of the intensity of coupling. We observe that St decreases monotonically with ϕ_m for all St_0 considered, the strongest reduction being of the order of 40%. As a straightforward consequence and since $St_0 \gtrsim 1$, this effect goes together with an increase of particle concentration. A reduction of St indeed brings the distribution of the particles closer to the maximum of clustering, which is known to occur at $St \sim 1$.

In order to test this prediction, we report in Fig. 11(a) the measurement of the probability density function of the particle spatial density n_p for different intensities of the back-reaction and for the largest available Stokes number, *i.e.* $St \simeq 17$. In the absence of coupling, that is when $\phi_\mathrm{m} = 0$, the distribution is very narrow around its average $\langle n_\mathrm{p}\rangle = n_0 = 1/L^2$, which, by conservation of mass and spatial homogeneity of the statistics, is independent of both ϕ_m and St. The observed narrow distribution indicates that the spatial repartition of the mass is nearly uniform, as it is expected for particles with large Stokes numbers. This is in agreement with the observed instantaneous density field displayed in Fig. 2 (lower left panel). When the coupling is turned on, that is when $\phi_\mathrm{m} > 0$, the distribution significantly broadens. Regions with very few or with high particle-number become more probable, so that clustering is enhanced. We note that this effect is much stronger than what would be expected by the sole reduction of the Stokes number. One notice that the distribution of particle density develops a power-law tail at small values $n_\mathrm{p} \ll n_0$. The exponent associated to this decreases when the intensity of coupling increases. It reveals an increasing contribution of voids in the statistics of the particle spatial density.

The intensity of clustering can be quantified by the variance $\langle(n_\mathrm{p} - n_0)^2\rangle$ of density fluctuations. This quantity is represented as a function of ϕ_m in Fig. 11(b) for the various reference Stokes numbers St_0 considered in this work. The error bars are computed as follows: for every time series of the second order moment of the field $n_p(\boldsymbol{x}, t)$, we compute a sliding mean $m(t) = \langle\langle n_p^2(\boldsymbol{x}, t)\rangle_{t\in[T_0,T]}\rangle_{\boldsymbol{x}}$ with an increasing windowing size $\tau = T - T_0$. T_0 is the time at which the distribution reaches a statistically stationary state. A time

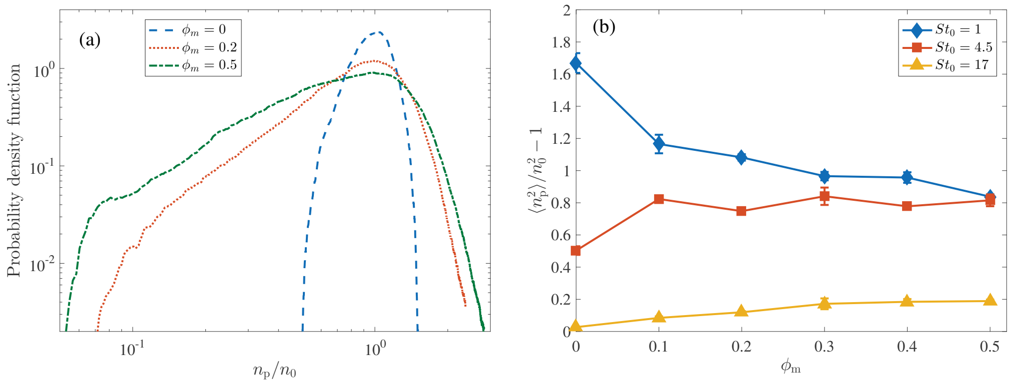


FIGURE 11. (a) Probability density function of particle density $\rho_p(\boldsymbol{x})$ represented for different $\phi_{\rm m}$ and $St_0 = 17$. Note that $\langle \rho_p \rangle = 1/L^2$ is independent of $\phi_{\rm m}$. (b) Variance of the density fluctuations as a function of $\phi_{\rm m}$ for $St_0 \sim 17$.

$t = T^*$ is chosen at which the average $m(t)$ seems to be stationary. The error is estimated as $\max_{t\in[T^*,T_{\max}]} |m(t) - m(T_{\max})|$. For the highest Stokes number (bottom line with triangular symbols) and as pointed out earlier, those fluctuations indeed increase as a function of the mass loading $\phi_{\rm m}$. This is also true for the intermediate Stokes number. However, at the lowest value, the behavior changes. We indeed observe for $St_0 \simeq 1$ a decrease of clustering when the coupling intensity increases. This effect can be partially explained by the reduction of the Stokes number which brings the particles away from the maximum of clustering. Another effect which can contribute to the same phenomenon may be due to small scale instabilities, which occur at small Stokes and that were already mentioned in §4. These instabilities are responsible for disrupting the large-scale vortices of the flow from which the particles are ejected. This mechanism thus leads to an increase of particle mixing.

7. Summary and conclusion

We have investigated in this work the modulation of turbulence by heavy, point-like particles in the two-dimensional direct enstrophy cascade. We made use of a field formulation in position-velocity phase space for the particle dynamics, following the recent development in Laenen *et al.* (2016) of a numerical method whose validity has been assessed at large Stokes numbers. The fluid flow being characterized by a unique timescale, particles dynamics in the direct enstrophy cascade is fully characterized by a unique Stokes number. At large values of the Stokes number, we found that the effects of two-way coupling are fully reproduced by an effective Ekman damping associated to an extra linear friction term in Navier–Stokes equation. This purely dissipative effect disappear at Stokes numbers of the order of unity or smaller. The particle phase still acts as an enstrophy pump at large scales but is able to transfer and re-inject it at smaller scales. As a result of this, the non-linear transfer over scales is reversed leading to a deep qualitative change of the direct cascade phenomenology.

We also observed in this work a marked effect of coupling on the intermittency of the fluid flow. In the scaling range, this effect is actually a straightforward consequence of the modulation of second-order statistics. We have indeed seen that the scaling exponents of the vorticity structure functions collapse to a universal curve, once they are rescaled by the second-order exponent, and independently of either the particle mass loading or the coefficient of Ekman damping. At small scales, the coupling with a particle phase is responsible for a flattening of the probability density function of vorticity gradients. Finally, it was shown that the two-way coupling also impacts clustering. The distribution of the particle spatial density field develops larger tails as the coupling intensity increases. This effect might be partly attributed to the reduction of particles relative inertia due to the attenuation of enstrophy fluctuations in the flow.

A clear extension of this work could consist in repeating such a study in the case of the inverse energy cascade. In that case, characteristic timescales τ_r depend on the observation scale r and monotonically increases with it as $r^{2/3}$. Hence, the strength of particle inertia is measured by a scale-dependent, local Stokes number $St(r) = \tau_p/\tau_r$. When the scale associated to $St(r) = 1$ falls inside the inertial range, the small scales are impacted by particles with $St < 1$, while large scales feel particles with $St > 1$. This would lead to an intricate situation where both phenomenologies are present. Such settings might clearly shed lights on what could happen in a fully developed, three-dimensional turbulent flow.

This work benefited from useful discussions with Giorgio Krstulovic who is warmly acknowledged. This research has received funding from the French Agence Nationale de la Recherche (Programme Blanc ANR- 12-BS09-011-04). Simulations were performed using HPC resources from the Mesocenter SIGAMM hosted by the Observatoire de la Côte dAzur.

Appendix A. Energy conservation and dissipative effect of particles

We derive here a prediction for the sign of ε_C which is the contribution in the fluid kinetic energy budget coming from the interaction with the particles — see equation (4.1). We denote by $\mathcal{E}_\mathrm{f}(\boldsymbol{x},t) = |\boldsymbol{u}(\boldsymbol{x},t)|^2/2$ the spatial fluid kinetic energy field, $\mathcal{E}_\mathrm{p}(\boldsymbol{x},t) = \int f(\boldsymbol{x},\boldsymbol{w},t)|\boldsymbol{w}|^2\,\mathrm{d}^2w/2$ that of the particle phase and $E_\mathrm{p} = \langle \mathcal{E}_\mathrm{p}(\boldsymbol{x},t)\rangle$ the average energy of the particles. Here and in the following the average is over time and the spatial domain.

We first write down the definition of ε_C:

$$\varepsilon_C = -\frac{1}{\tau_\mathrm{p}}\left[\langle n_\mathrm{p}\boldsymbol{v}_\mathrm{p}\cdot\boldsymbol{u}\rangle - \langle n_\mathrm{p}|\boldsymbol{u}|^2\rangle\right] \tag{A 1}$$

Using the Liouville equation (2.2), the conservation of particle kinetic energy reads:

$$\frac{\mathrm{d}E_\mathrm{p}}{dt} = \langle \int \partial_t f\,|\boldsymbol{w}|^2\,\mathrm{d}^2w\rangle = \frac{1}{\tau_\mathrm{p}}\left[\langle n_\mathrm{p}\boldsymbol{v}_\mathrm{p}\cdot\boldsymbol{u}\rangle - 2\langle\mathcal{E}_\mathrm{p}\rangle\right] \tag{A 2}$$

Furthermore, using Holder inequality, it can be shown that

$$\int f|\boldsymbol{w}|^2\,\mathrm{d}^2w \times \int f\,\mathrm{d}^2w \geqslant \left|\int f\boldsymbol{w}\,\mathrm{d}^2w\right|^2, \quad \text{so that } 2\mathcal{E}_\mathrm{p} \geqslant n_\mathrm{p}|\boldsymbol{v}_\mathrm{p}|^2 \tag{A 3}$$

Since $\mathrm{d}E_\mathrm{p}/\mathrm{d}t \geqslant 0$ (particles are initially at rest), equation (A 2) implies

$$\langle n_\mathrm{p}\boldsymbol{v}_\mathrm{p}\cdot\boldsymbol{u}\rangle \geqslant 2\langle\mathcal{E}_\mathrm{p}\rangle \geqslant \langle n_\mathrm{p}|\boldsymbol{v}_\mathrm{p}|^2\rangle \tag{A 4}$$

We further notice that $\langle n_p\boldsymbol{v}_\mathrm{p}\cdot\boldsymbol{u}\rangle$ cannot be simultaneously greater than $\langle n_\mathrm{p}|\boldsymbol{u}|^2\rangle$ and $\langle n_\mathrm{p}|\boldsymbol{v}_\mathrm{p}|^2\rangle$ because of the Cauchy-Schwartz inequality. Hence $\langle n_\mathrm{p}|\boldsymbol{u}|^2\rangle \geqslant \langle n_p\boldsymbol{v}_\mathrm{p}\cdot\boldsymbol{u}\rangle$ and $\varepsilon_C \geqslant 0$, meaning that two-way coupling may only withdraw kinetic energy from the fluid.

REFERENCES

Balachandar, S. 2009 A scaling analysis for point-particle approaches to turbulent multiphase flows. *Int. J. Multiphase Flow* **35** (9), 801–810.

Bec, J., Ray, S.S., Saw, E.-W. & Homann, H. 2016 Abrupt growth of large aggregates by correlated coalescences in turbulent flow. *Phys. Rev. E* **93** (3), 031102.

Bernard, D. 2000 Influence of friction on the direct cascade of the 2d forced turbulence. *Europhys. Lett.* **50** (3), 333–339.

Boffetta, G., Celani, A., Musacchio, S. & Vergassola, M. 2002 Intermittency in two-dimensional Ekman-Navier-Stokes turbulence. *Phys. Rev. E* **66** (2), 026304.

Boffetta, G., Cenedese, A., Espa, S. & Musacchio, S. 2005 Effects of friction on 2D turbulence: An experimental study of the direct cascade. *Europhys. Lett.* **71** (4), 590–596.

Boffetta, G. & Ecke, R.E. 2012 Two-dimensional turbulence. *Annu. Rev. Fluid Mech.* **44**, 427–451.

Cisse, M., Homann, H. & Bec, J. 2013 Slipping motion of large neutrally buoyant particles in turbulence. *J. Fluid Mech.* **735**, R1.

Cisse, M., Saw, E.-W., Gibert, M., Bodenschatz, E. & Bec, J. 2015 Turbulence attenuation by large neutrally buoyant particles. *Phys. Fluids* **27** (6), 061702.

Crowe, C.T., Sharma, M.P. & Stock, D.E. 1977 The Particle-Source-In Cell (PSI-CELL) Model for Gas-Droplet Flows. *J. Fluids Eng.* **99** (2), 325–332.

Curtis, J.S. & Van Wachem, B. 2004 Modeling particle-laden flows: A research outlook. *AIChE J.* **50** (11), 2638–2645.

Danilov, S.D. & Gurarie, D. 2000 Quasi-two-dimensional turbulence. *Physics-Uspekhi* **43** (9), 863–900.

Durham, W.M., Climent, E., Barry, M., De Lillo, F., Boffetta, G.and Cencini, M. & Stocker, R. 2013 Turbulence drives microscale patches of motile phytoplankton. *Nature Comm.* **4**.

Elghobashi, S 1994 On predicting particle-laden turbulent flows. *Appl. Sci. Res.* **52** (4), 309–329.

Falkovich, G., Fouxon, A. & Stepanov, M.G. 2002 Acceleration of rain initiation by cloud turbulence. *Nature* **419** (6903), 151–154.

Gao, H., Li, H. & Wang, L.-P. 2013 Lattice Boltzmann simulation of turbulent flow laden with finite-size particles. *Comput. Math. Appl.* **65** (2), 194–210.

Garg, R., Narayanan, C. & Subramaniam, S. 2009 A numerically convergent Lagrangian Eulerian simulation method for dispersed two-phase flows. *Int. J. Multiphase Flow* **35** (4), 376–388.

Gualtieri, P., Picano, F., Sardina, G. & Casciola, C.-M. 2011 Clustering and turbulence modulation in particle laden shear flows. *J. Phys.: Conf. Series* **333**, 012007–17.

Gualtieri, P., Picano, F., Sardina, G. & Casciola, C.-M. 2015 Exact regularized point particle method for multiphase flows in the two-way coupling regime. *J. Fluid Mech.* **773**, 520–561.

Gustavsson, K., Mehlig, B., Wilkinson, M. & Uski, V. 2008 Variable-range projection model for turbulence-driven collisions. *Phys. Rev. Lett.* **101** (17), 174503.

Gyr, A. & Rys, F.-S. 2013 *Diffusion and transport of pollutants in atmospheric mesoscale flow fields*, , vol. 1. Springer Science & Business Media.

Hwang, W. & Eaton, J.K. 2006 Homogeneous and isotropic turbulence modulation by small heavy ($st \sim 50$) particles. *J. Fluid Mech.* **564**, 361–393.

Jacquet, E., Balbus, St. & Latter, H. 2011 On linear dust–gas streaming instabilities in protoplanetary discs. *Month. Not. Roy. Astron. Soc.* **415** (4), 3591–3598.

Jenny, P., Roekaerts, D. & Beishuizen, N. 2012 Modeling of turbulent dilute spray combustion. *Prog. En. Combust. Sc.* **38** (6), 846–887.

Johansen, A. & Youdin, A. 2007 Protoplanetary disk turbulence driven by the streaming instability: nonlinear saturation and particle concentration. *Astrophys. J.* **662** (1), 627.

Kraichnan, R.H. 1967 Inertial ranges in two-dimensional turbulence. *Phys. Fluids* **10**, 1417–1423.

Laenen, F., Krstulovic, G. & Bec, J. 2016 A lattice method for the Eulerian simulation of heavy particle suspensions. *C.R. Mecanique* **344**, 672–683.

Lanotte, A.S., Seminara, A. & Toschi, F. 2009 Cloud droplet growth by condensation in homogeneous isotropic turbulence. *J. Atmo. Sci.* **66** (6), 1685–1697.

LeVeque, R.J. 2002 *Finite volume methods for hyperbolic problems*. Cambridge university press.

Li, J., Wang, H., Liu, Z., Chen, S. & Zheng, C. 2012 An experimental study on turbulence modification in the near-wall boundary layer of a dilute gas-particle channel flow. *Exp. Fluids* **53** (5), 1385–1403.

Loisy, A. & Naso, A. 2016 The interaction between a large buoyant bubble and turbulence. https://hal.archives-ouvertes.fr/hal-01346796/.

Lucci, F., Ferrante, A. & Elghobashi, S. 2010 Modulation of isotropic turbulence by particles of Taylor length-scale size. *J. Fluid Mech.* **650**, 5–55.

Mandø, M. 2009 Turbulence modulation by non-spherical particles. PhD thesis, Aalborg Universitet.

Matas, J.-P., Morris, J. F. & Guazzelli, É. 2003 Transition to turbulence in particulate pipe flow. *Phys. Rev. Lett.* **90**, 014501.

Maxey, M.R. & Riley, J.J. 1983 Equation of motion for a small rigid sphere in a nonuniform flow. *Phys. Fluids* **26** (4), 883–889.

Nam, K., Ott, E., Antonsen Jr., T.A. & Guzdar, P.N. 2000 Lagrangian chaos and the effect of drag on the enstrophy cascade in two-dimensional turbulence. *Phys. Rev. Lett.* **84** (22), 5134–5137.

Patro, P. & Dash, S.K. 2014 Computations of particle-laden turbulent jet flows based on Eulerian model. *J. Fluids Eng.* **136** (1), 011301.

Perlekar, P. & Pandit, R. 2009 Statistically steady turbulence in thin films: direct numerical simulations with Ekman friction. *New J. Phys.* **11** (7), 073003.

Poesio, P., Ooms, G., Ten Cate, A. & Hunt, J.C.R. 2006 Interaction and collisions between particles in a linear shear flow near a wall at low reynolds number. *J. Fluid Mech.* **555**, 113–130.

Post, S.L. & Abraham, J. 2002 Modeling the outcome of drop–drop collisions in Diesel sprays. *Int. J. Multiphase Flow* **28** (6), 997–1019.

Sabelnikov, V., Ovsyannikov, A.Y. & Gorokhovski, M. 2014 Modified level set equation and its numerical assessment. *J. Comput. Phys.* **278**, 1–30.

Saber, A., Lundström, T.S & Hellström, J.G.I. 2015 Turbulent modulation in particulate flow: A review of critical variables. *Engineering* **7** (10), 597.

Succi, Sauro 2001 *The Lattice Boltzmann Equation for fluid dynamics and beyond*. Oxford University Press, USA.

Ten Cate, A., Derksen, J.J., Portela, L.M. & Van Den Akker, H.E.A. 2004 Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence. *J. Fluid Mech* **519**, 233–271.

Verma, M.K. 2012 Variable enstrophy flux and energy spectrum in two-dimensional turbulence with Ekman friction. *Europhys. Lett.* **98** (1), 14003–7.

Yang, T.-S. & Shy, S.S. 2005 Two-way interaction between solid particles and homogeneous air turbulence: particle settling rate and turbulence modification measurements. *J. Fluid Mech.* **526**, 171–216.

Dusty turbulence

Jérémie Bec,[1] François Laenen,[1] and Stefano Musacchio[2]
[1] *Université Côte d'Azur, CNRS, OCA, Laboratoire J.-L. Lagrange, Nice, France*
[2] *Université Côte d'Azur, CNRS, Laboratoire J.-A. Dieudonné, Nice, France*

The feedback foces exerted by particles suspended in a turbulent flow is shown to lead to a new scaling law for velocity fluctuations associated to a power-spectra $\propto k^{-2}$. The mechanism at play relies on a direct transfer of kinetic energy to small scales through Kelvin–Helmholtz instabilities occurring in regions of high particle density contrast. This finding is confirmed by two-dimensional direct numerical simulations.

It is common to face environmental, industrial or astrophysical situations where impurities such as dust, droplets, sediments, and other kinds of colloids are transported by a turbulent fluid. When the suspended particles have finite sizes and masses, they detach from the flow by inertia and form uneven distributions where intricate interactions and collisions take place. The physical processes at play are rather well established, leading to quantitative predictions on the rates at which cloud droplets coalesce [1], dust accrete to form planets [2], or heavy sediments settle in a turbulent environment [3, 4].

Still, basic and important questions remain largely open as to the backward influence of particles on the carrier flow structure and geometry. Some situations involve particle mass loadings so large that the fluid turbulent microscales are altered and, in turn, several macroscopic processes are drastically impacted. These include spray combustion in engines [5], aerosol saltation in dust storms [6], biomixing by microorganisms in the oceans [7], and formation of planetesimals by streaming instabilities in circumstellar disks [8]. Currently such systems are unsatisfactorily handled by empirical approaches or specific treatments. A better modelling requires identifying and understanding the *universal* physical mechanisms at play in turbulence modulation by dispersed particles. In this spirit, we focus here on the alteration of small scales by tiny heavy spherical particles. We show that the fluid velocity is unstable in regions with a high particle density contrast, leading to energy transfers shortcutting the classical turbulent cascade. This effect leads to a novel scaling regime of the turbulent velocity field associated to a power-law spectrum $\propto k^{-2}$.

The fluid velocity field $\boldsymbol{u}$ solves the incompressible Navier–Stokes equations: $\nabla\cdot\boldsymbol{u}=0$ with

$$\partial_t\boldsymbol{u}+(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}=-\frac{1}{\rho_\mathrm{f}}\nabla p+\nu\,\nabla^2\boldsymbol{u}+\boldsymbol{f}_\mathrm{ext}+\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}. \quad (1)$$

ρ_f is here the fluid mass density and ν its kinematic viscosity. A homogeneous isotropic turbulence is maintained in a statistical steady state by an external forcing $\boldsymbol{f}_\mathrm{ext}$. The fluid flow is perturbed by a monodisperse population of small solid particles whose effects are entailed in the force $\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}$. These particles are assumed sufficiently small, dilute and heavy for approximating their distribution and dynamics in terms of fields, namely a mass density ρ_p and a particle velocity field $\boldsymbol{v}_\mathrm{p}$ satisfying

$$\partial_t\rho_\mathrm{p}+\nabla\cdot(\rho_\mathrm{p}\boldsymbol{v}_\mathrm{p})=0 \quad (2)$$

$$\partial_t\boldsymbol{v}_\mathrm{p}+(\boldsymbol{v}_\mathrm{p}\cdot\nabla)\,\boldsymbol{v}_\mathrm{p}=-\frac{1}{\tau_p}\,(\boldsymbol{v}_\mathrm{p}-\boldsymbol{u}), \quad (3)$$

where $\tau_\mathrm{p}=2\rho_\mathrm{s}\,a^2/(9\rho_\mathrm{f}\,\nu)$ is the particles response time, a being their radius and ρ_s the mass density of the material constituting the particles. The hydrodynamical system (2)-(3) has proven to be a valid approximation for relatively small Stokes numbers $St=\tau_\mathrm{p}/\tau_\mathrm{f}$ [9], that is when the particle response time is smaller than the smallest active timescale τ_f of the fluid flow. In this limit, fold caustics appear with an exponentially small probability [10, 11], preventing the development of multivalued branches in the particle velocity profile and thus ensuring the validity of a hydrodynamical description.

The force exerted by the particles on the fluid reads

$$\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}=\frac{1}{\tau_\mathrm{p}}\,\frac{\rho_\mathrm{p}}{\rho_\mathrm{f}}\,(\boldsymbol{v}_\mathrm{p}-\boldsymbol{u})\,. \quad (4)$$

It is proportional to the mass density of the dispersed phase and thus combines the heaviness of the particles with their number density. The strength of feedback is measured by the comprehensive non-dimensional parameter $\Phi=\langle\rho_\mathrm{p}\rangle/\rho_\mathrm{f}$. It involves the particle density spatial average $\langle\rho_\mathrm{p}\rangle=N_\mathrm{p}\,m_\mathrm{p}/\mathcal{V}$, where N_p is the total number of particles, m_p their individual mass, and $\mathcal{V}$ the volume of the domain. All these quantities being conserved by the dynamics, so is the coupling parameter Φ.

We first draw some straightforward comments pertaining to the limit of small Stokes numbers. There, particles almost follow the flow with a tiny compressible correction [12], namely $\boldsymbol{v}_\mathrm{p}\approx\boldsymbol{u}-\tau_\mathrm{p}\,\boldsymbol{a}$, where $\boldsymbol{a}=\partial_t\boldsymbol{u}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}$ denotes the fluid flow acceleration field. The feedback force exerted on the fluid is hence, to leading order,

$$\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}(\boldsymbol{x},t)\approx-\frac{1}{\rho_\mathrm{f}}\,\rho_\mathrm{p}(\boldsymbol{x},t)\,\boldsymbol{a}(\boldsymbol{x},t) \quad (5)$$

The effect of particles can thus be seen as an *added mass*, which does not depend upon their response time and is responsible for an increase of the fluid inertia. The fluid is accelerated as if it has an added density equal to that of the particles. Such considerations predict that the presence of particles decreases the effective kinematic viscosity of the fluid and thus increases its level of turbulence.

This vision is however too naive as it overlooks the spatial fluctuations of the particle density. It is indeed known that an even infinitesimal inertia of the particles creates extremely violent gradients of their density through the mechanism of preferential concentration. As we will now see, these variations are responsible for instabilities that shortcut the turbulent energy cascade by directly transferring kinetic energy to the smallest turbulent scales.

A key attribute of turbulence is the vigorous local spinning of the fluid flow, weighed by the vorticity $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$. The effect of particles on the vorticity dynamics is entailed in the curl of the feedback force (4), reading

$$\nabla \times \boldsymbol{f}_{\mathrm{p}\to\mathrm{f}} = \frac{1}{\rho_\mathrm{f}\,\tau_\mathrm{p}}\left[\rho_\mathrm{p}\left(\boldsymbol{\omega}_\mathrm{p}-\boldsymbol{\omega}\right) + \nabla\rho_\mathrm{p}\times\left(\boldsymbol{v}_\mathrm{p}-\boldsymbol{u}\right)\right], \tag{6}$$

where $\boldsymbol{\omega}_\mathrm{p} = \nabla\times\boldsymbol{v}_\mathrm{p}$ is the vorticity of the dispersed phase. The action of particles is thus twofold. The first term accounts for a friction of the fluid vorticity with that of the particles, which amounts at small Stokes numbers to the above-mentioned added-mass effect. The second term gives a source of vorticity proportional to the gradients of particle density. The combined effects of preferential concentration and turbulent mixing is responsible for very sharp spatial variations of ρ_p. Centrifugal forces indeed eject heavy particles from coherent vortical structures [13] and Lagrangian transport stretches particles patches in stirring regions [14]. This leads to the development of substantial fluctuations of $\nabla\rho_\mathrm{p}$, as illustrated in two dimensions on the left panels of Fig. 1. This mechanism creates regions with very strong shear in the fluid flow, which, in turn, develop small-scale vortical structures through Kelvin–Helmholtz instability. It is indeed well known that flows presenting a quasi-discontinuity of velocity are linearly unstable and develop wavy vortical streaks at the interface of the two motions (see, *e.g.*, [15]). Such phenomenological arguments thus suggest that the feedback of particles lead to the formation of small-scale eddies, as can be seen in the right panels of Fig. 1. Particles thus actively participate in the transfer of kinetic energy toward the smallest turbulent scales.

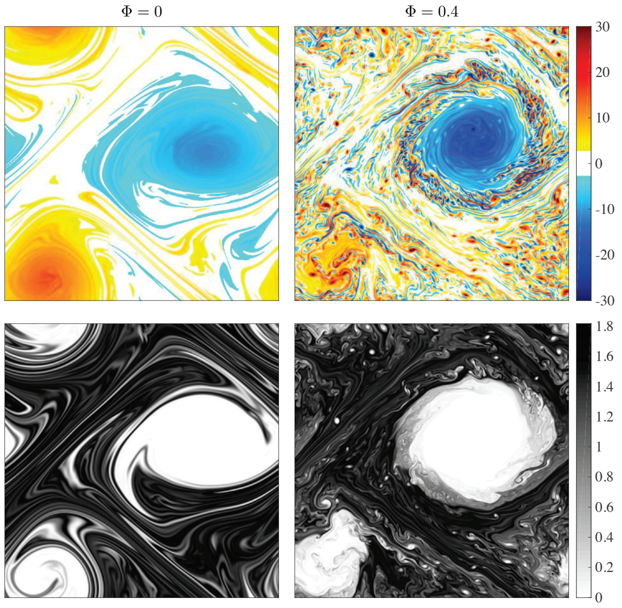


FIG. 1. (color online) Upper frames: Snapshots of the (scalar) fluid vorticity $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$ in two dimensions for both the passive case with no feedback ($\Phi = \langle\rho_\mathrm{p}\rangle/\rho_\mathrm{f} = 0$, left) and when the particles exert a back reaction on the flow ($\Phi = 0.4$, right). Lower frames: normalized particle density $\rho_\mathrm{p}/\langle\rho_\mathrm{p}\rangle$ at the same instants of time. The two cases (without and with coupling) correspond to different realizations of the external forcing; the x and y axis were however shifted in order to locate large-scale structures at approximately the same position.

This effect and the resulting modification of the fluid flow scaling properties can be quantified by examining the scale-by-scale kinetic energy budget given by Kármán–Howarth–Monin relation (see, *e.g.*, [16]). Denoting the velocity increment over a separation $\boldsymbol{r}$ by $\delta_r\boldsymbol{u} = \boldsymbol{u}' - \boldsymbol{u}$ with $\boldsymbol{u}' = \boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r},t)$ and $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x},t)$, one can easily check that statistically homogeneous solutions to the Navier–Stokes equation (1) satisfy

$$\begin{aligned}\frac{1}{2}\partial_t\left\langle\boldsymbol{u}\cdot\boldsymbol{u}'\right\rangle = {} & \frac{1}{4}\nabla_r\cdot\left\langle\left|\delta_r\boldsymbol{u}\right|^2\delta_r\boldsymbol{u}\right\rangle + \nu\,\nabla_r^2\left\langle\boldsymbol{u}\cdot\boldsymbol{u}'\right\rangle \\ & + \left\langle\boldsymbol{u}\cdot\bar{\boldsymbol{f}}_\mathrm{ext}\right\rangle + \left\langle\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}\cdot\bar{\boldsymbol{u}}\right\rangle,\end{aligned} \tag{7}$$

where the overbar denotes the average over the two points located at $\boldsymbol{x}\pm\boldsymbol{r}$, that is $\bar{\boldsymbol{f}} = [\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{r},t)+\boldsymbol{f}(\boldsymbol{x}-\boldsymbol{r},t)]/2$. In classical stationary turbulence, the above relation suggests a balance between the non-linear transfer term (first term on the right-hand side) and viscous dissipation (second term), leading for isotropic flows to the celebrated Kolmogorov 4/5 law. In the presence of coupling with particles, this equilibrium is broken by the feedback force. In the asymptotics $St \ll 1$ of low inertia, this force is approximated by (5), so that its contribution to (7) reads

$$\left\langle\boldsymbol{f}_{\mathrm{p}\to\mathrm{f}}\cdot\bar{\boldsymbol{u}}\right\rangle \approx \frac{1}{\rho_\mathrm{f}}\left\langle\rho_\mathrm{p}\,\boldsymbol{a}\cdot\boldsymbol{u}\right\rangle - \frac{1}{\rho_\mathrm{f}}\left\langle\rho_\mathrm{p}\,\boldsymbol{a}\cdot\delta_r\boldsymbol{u}\right\rangle. \tag{8}$$

The first term on the right-hand side involves the correlation between the particle density field and the instantaneous power acting on fluid elements. To leading order when $St\to 0$, we have $\langle\rho_\mathrm{p}\,\boldsymbol{a}\cdot\boldsymbol{u}\rangle \approx \langle\rho_\mathrm{p}\rangle\,\langle\boldsymbol{a}\cdot\boldsymbol{u}\rangle = 0$. Non-vanishing corrections at small but finite Stokes numbers might arise from a combined effect of the small compressibility of the particle velocity together with the biased sampling due to preferential concentration, as already seen for the radial distribution function [17, 18]. However such correlations are in the best case of the order of St^2. The second term on the right-hand side of (8) does not vanish in the limit $St\to 0$ and thus gives the dominant contribution.

Such arguments lead to predict that the scale-by-scale energy balance (7) reduces in the inertial range to

$$\frac{1}{4}\nabla_r \cdot \left\langle |\delta_r \boldsymbol{u}|^2\, \delta_r \boldsymbol{u} \right\rangle \simeq \frac{1}{\rho_{\rm f}} \left\langle \rho_{\rm p}\, \boldsymbol{a} \cdot \delta_r \boldsymbol{u} \right\rangle . \tag{9}$$

Now, assuming that the fluid velocity field obeys some scaling property $\delta_r \boldsymbol{u} \sim r^h$, one deduces from the above balance that $3h - 1 = h$, and thus $h = 1/2$. Such a scaling behavior is associated to an angle-averaged kinetic energy power spectrum $E(k) \propto k^{-2}$.

In order to test such prediction, we perform two-dimensional simulations of the fluid-particle system defined by (1), (2), (3), and (4) in a periodic domain. We make use of a Fourier-spectral solver with 1024^2 collocation points for estimating spatial derivatives and of a second-order Runge-Kutta scheme for time marching. We focus on the direct enstrophy cascade, so that the external forcing $\boldsymbol{f}_{\rm ext}$ is the sum of an Ekman friction with timescale $1/\alpha$ and of a random Gaussian field η white noise in time and concentrated at wavenumbers $|\boldsymbol{k}| \leq 2$. We make use of hyper-viscosity and hyper-diffusivity (fourth power of the Laplacian) in order to maximize the extent of the inertial range and prevent Eqs. (2) and (3) from blowing up. The particle response time is fixed in such a way that $St = \tau_{\rm p} \langle \omega^2 \rangle^{1/2} \approx 10^{-2}$ in the uncoupled case and various values of the coupling parameter $\Phi = 0,\ 0.1,\ 0.2$, and 0.4 are simulated.

Figure 1 shows snapshots of the fluid vorticity field together with the particle density field, without and with coupling between the two phases. In the absence of feedback from the particles (left panels), the flow develops the traditional picture of two-dimensional direct cascade consisting of large-scale vortices separated by a bath of filamentary structures where enstrophy is dissipated. The particles density field is characterized by large voids in the vortical structures separated by a filamentary distribution that is symptomatic of turbulent mixing. These qualitative pictures are strongly altered when the particle feedback is turned on. In the presence of coupling (right panels), the fluid flow still shows large-scale structures but which are this time surrounded by a bath of small-scale vortices. These eddies form wavy structures along the lines associated to quasi-discontinuities of the particle density field. This is a clear signature that Kelvin–Helmholtz instability is at play.

Figure 2 shows the angle-averaged power spectra of the fluid kinetic energy obtained when varying the coupling parameter. In the case of no feedback ($\Phi = 0$), the specific choices of the Ekman coefficient α and of the energy injection amplitude yield a kinetic energy spectrum $E(k) \propto k^{-\delta}$ with $\delta \approx 3.3$. For any non-vanishing value of the coupling parameter Φ, one observes remarkable changes in the spectral behavior of the fluid velocity. The first effect is a clear decrease of the total kinetic energy. Similarly to what is obtained in the asymptotic of large Stokes numbers [19], this is due to a net dissipative effect of the coupling with the particle phase. However this impacts only the largest scales of the flow and the smaller scales experience an increase in their energy content. The inertial-range is characterized by a shallower power spectrum with an exponent close to -2, as expected from above arguments. Dissipative scales are shifted toward larger wavenumbers, as a consequence of the added-mass effect induced by particles which decreases the effective kinematic viscosity of the fluid loaded by particles.

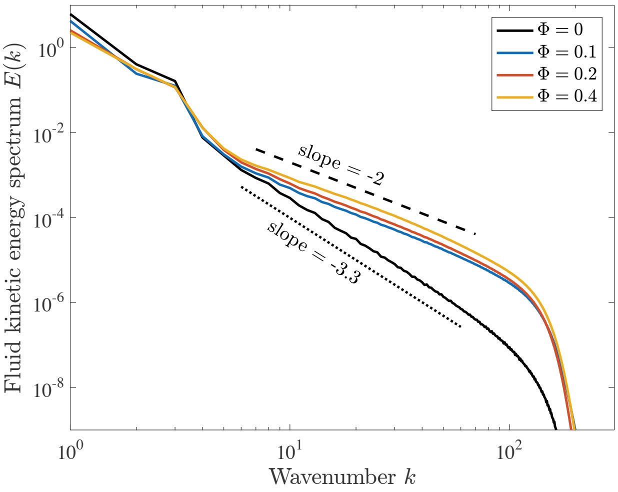


FIG. 2. (color online) Angle-averaged kinetic energy power spectra of the fluid velocity represented for various values of the coupling parameter Φ, as labelled.

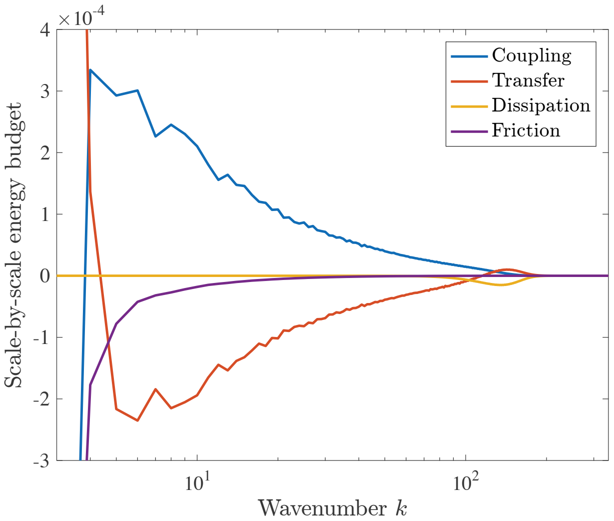


FIG. 3. (color online) Angle-averaged Fourier amplitudes of the various terms contributing to the kinetic energy budget (7) shown here for $\Phi = 0.4$. Coupling stands from the contribution of the forces exerted by the particles on the fluid, transfer for the nonlinear advection terms, dissipation for viscous forces and friction for Ekman damping.

Further insight is given by measuring the amplitude of the various terms entering in the energy budget (7). Figure 3 shows the angle-averaged amplitude of their Fourier transforms with respect to the separation $\boldsymbol{r}$. One observes that the non-linear transfer term gives a positive contribution at small wavenumbers. This is a strong signature of two-dimensional turbulence for which, conversely to three dimensions, the nonlinear terms are not transferring kinetic energy toward small scales participating to its accumulation at largest lengthscales of the flow. This term is exactly compensated by the linear Ekman friction and the coupling with the particle phase which are both negative and of the same order. Coupling is thus pumping energy at large scales but restitutes it at larger wavenumbers as it is positive for $k \geq 4$. In the inertial range for $10 \lesssim k \lesssim 100$ where both the contribution of Ekman friction and viscous dissipation are negligible, it is exactly compensated by a negative value of the nonlinear transfer term. Both curves decrease as k^{-1}, in agreement with the scaling observed earlier. At the smallest scales, coupling becomes negligible, nonlinear transfer changes sign and is compensated by viscous dissipation. The whole two-dimensional picture thus confirms the prediction made above.

We have thus evidenced from this work a new regime of turbulent flow where the feedback of suspended particles onto the fluid flow dominates inertial-range energy transfers. This regime is evidenced by numerical simulations in two dimensions but such strong effects should also be present in three dimensions, at least at sufficiently small scales. A remarkable feature of this turbulent enhancement due to dust-like particles is the creation of small-scale eddies whose spectral signature is a k^{-2} power-law range for the fluid velocity. These vortices profoundly affect particle concentration. On the one-hand, their spatial distribution tends to weaken large-scale inhomogeneities, to reduce potential barriers to transport and enhance mixing. On the other hand, the dispersion in the flow and the interactions between these long-living structures trigger density fluctuations that are much more intense than in the absence of coupling between the two phases. Such effects clearly need being investigated in a more systematic manner: They might indeed strongly modify at both qualitative and quantitative levels the rate at which particles interact together.

We acknowledge useful discussions with G. Krstulovic. The research leading to these results has received funding from the French Agence Nationale de la Recherche (Programme Blanc ANR-12-BS09-011-04).

[1] W. W. Grabowski and L.-P. Wang, Ann. Rev. Fluid Mech. **45**, 293 (2013).
[2] A. Johansen, M.-M. Mac Low, P. Lacerda, and M. Bizzarro, Science Advances **1**, e1500109 (2015).
[3] J. Bec, H. Homann, and S. S. Ray, Phys. Rev. Lett. **112**, 184501 (2014).
[4] K. Gustavsson, S. Vajedi, and B. Mehlig, Phys. Rev. Lett. **112**, 214501 (2014).
[5] P. Jenny, D. Roekaerts, and N. Beishuizen, Prog. Energy Combust. Sci. **38**, 846 (2012).
[6] J. F. Kok, E. J. Parteli, T. I. Michaels, and D. B. Karam, Rep. Prog. Phys. **75**, 106901 (2012).
[7] A. W. Visser, Science **316**, 838 (2007).
[8] A. Johansen, J. S. Oishi, M.-M. Mac Low, H. Klahr, T. Henning, and A. Youdin, Nature **448**, 1022 (2007).
[9] G. Boffetta, A. Celani, F. De Lillo, and S. Musacchio, Europhys. Lett. **78**, 14001 (2007).
[10] M. Wilkinson, B. Mehlig, and V. Bezuglyy, Phys. Rev. Lett. **97**, 048501 (2006).
[11] K. Gustavsson and B. Mehlig, Physical Review E **84**, 045304 (2011).
[12] M. R. Maxey, J. Fluid Mech. **174**, 441 (1987).
[13] M. R. Maxey, Phys. Fluids **30**, 1915 (1987).
[14] G. Haller and G. Yuan, Physica D **147**, 352 (2000).
[15] S. Chandrasekhar, *Hydrodynamic and Hydrodynamic stability* (Oxford University Press, 1961).
[16] U. Frisch, *Turbulence: the legacy of A.N. Kolmogorov* (Cambridge University Press, 1995).
[17] E. Balkovsky, G. Falkovich, and A. Fouxon, Phys. Rev. Lett. **86**, 2790 (2001).
[18] J. Chun, D. L. Koch, S. L. Rani, A. Ahluwalia, and L. R. Collins, J. Fluid Mech. **536**, 219 (2005).
[19] F. Laenen, S. Musacchio, and J. Bec, "Modulation of two-dimensional turbulence by small heavy particles," (2017), arXiv xxx.

CHAPTER 6

Conclusions and perspectives

This thesis explored several problems related to the transport of particles by turbulent flows. While such problems occur in many natural and industrial situations ranging from combustion engines to planet formation, predictive models are mainly based on eddy-diffusivity approaches and cannot accurately handle high concentration fluctuations or non-trivial feedback of the dispersed phase on the fluid flow. We have focused here on these two aspects: The first part of this thesis was dedicated to the dispersion by turbulence of tracers continuously emitted from a point source. The second part concerned the introduction of a novel numerical method to simulate the transport of inertial particles and to understand how they modulate the carrier turbulent flow.

6.1 Turbulent transport of particles emitted from a point source

In chapter 3, the emission of tracers from a point source in the two-dimensional inverse cascade was studied. The main issue characterizing such a system is that, even if particles are released from the same spatial location, they enter the domain at different times. As a consequence, quantifying turbulent mixing requires understanding both the temporal and spatial correlations of the flow. Numerical simulations of the two-dimensional inverse turbulent cascade have been performed. Lagrangian tracers were seeded from a fixed point in space into an Eulerian velocity field integrated by pseudo-spectral methods. An analysis was carried of the mean displacement of tracers, as well as of the average radial concentration profile. This allowed to conclude that the time-averaged dispersion is well described by two successive phases: A ballistic motion from the source with a characteristic velocity given by the large-scale flow, followed by diffusion after a time equal to the Lagrangian velocity correlation time.

Furthermore, a scaling analysis of the quasi-Lagrangian mass m_{QL} as a function of the distance from the source has been performed. To this end, the fractal correlation dimension was first measured using a classical box-counting algorithm at high spatial resolution $N_x^2 = 4096^2$. The signature of the linear particle distribution, due to the injected mechanism, was found to persist at small scales. This is in qualitative agreement with the persistence of the inhomogeneity (due to the injection mechanism) at small scales as it was concluded in Celani *et al.* (2007) in the case of the Kraichnan model for the passive scalar. However, the correlation dimension of the tracer distribution showed to be largely contaminated by the recurrence of the Wiener process in two dimensions. Long-living trajectories indeed come back arbitrarily close to the source and infinitely often, contributing a uniform background to the concentration of tracers and thus acting as a source of homogenisation. This effect was more and more contaminating the statistics as the particles maximum lifetime was increased.

To circumvent such difficulties, an alternative way to measure concentration fluctuations was further presented. This approach is based on a novel phenomenological description, which exploits the distortion of the emitted line as a function of time in order to quantify its contribution to the mass contained in balls of given size r and at a given distance R from the source. Although this approach is strictly valid in the limit of a continuous emitted line, some difficulties are encountered when working with discrete trajectories. In Lagrangian simulations, the continuity condition is quickly broken and we have shown that this effect limits the scaling analysis to a minimum size r that, in turn, depends on R.

A natural extension of this work could consist in relating the quasi-Lagrangian mass scaling to the two-point equal-time correlation function. A comparison of this quantity in the case of the Kraichnan velocity ensemble (Celani *et al.*, 2007) would allow one to stress the impact of a finite correlation time of the velocity. Also, as the Wiener process is not recurrent in dimension three and higher, it is expected that the determination of the correlation dimension $\mathcal{D}_2$ will not suffer from the homogenisation issue that has been encountered in two dimensions. It would be interesting to display the correspondences between the quasi-Lagrangian mass scaling obtained with the determination of $\mathcal{D}_2$ and the method we have presented by performing simulations in the three-dimensional direct cascade.

Another point that would be worth studying further is the universal character of the mass scaling. Indeed, the Richardson super-diffusive regime is a consequence of the non-smoothness of the velocity in the inertial range. For a kinetic energy spectrum $E(k) \propto k^{-\alpha}$, we have $\langle r^2 \rangle \propto t^{4/(3-\alpha)}$. Other types of turbulence display the same scaling $E(k) \propto k^{-5/3}$, such as surface quasi-geostrophic turbulence (SQG), or *single-layer* QG model. This kind of turbulence is for example observed at ocean surface (Lapeyre & Klein, 2006) and upper troposphere (Tulloch & Smith, 2006).

6.2 Modelisation of small inertial particles

In chapter 4, a numerical method has been proposed in order to resolve the kinetic equation associated with the dynamics of small inertial particles. This method allows to recover an Eulerian field for the particle density even at high inertia when the particle velocity dispersion has to be taken into account. Simulations were carried along with Lagrangian particles to assess for the validity of the method.

In a one-dimensional random flow, the ability of the method to recover the phase-space fractal distribution was assessed through the study of the radial distribution function. In two dimensions, it was shown that this approach is able to reproduce the spatial distribution of the particles and their centrifugal ejection outside the vortical structures of the flow. Such preferential concentration effects were tested in two-dimensional turbulence, both in direct and inverse cascade, as well as in a cellular flow. The numerical convergence of the proposed method was studied via the analysis of the numerical error with respect to the velocity resolution. Also, the numerical performance was compared to Lagrangian simulations, showing an advantage for the Eulerian formulation as long as moderate errors $\mathcal{O}(10^{-2})$ on the density are tolerable.

Among questions that remain open, let us mention possible improvements of this method. For example, one can wonder whether or not the designed approach would be adaptable to polydisperse suspensions. Up to now it was used to integrate numerically the Liouville equation for the particle density in the position-velocity phase space $(\boldsymbol{x}, \boldsymbol{u})$, focusing on particles that are characterised by a unique Stokes number quantifying their inertia. A naive way to consider N particle phases (of different sizes or masses) would be to implement N different density fields f. This would obviously multiply the computational complexity by N. This would open the way to consider interactions exchanging mass and momentum between the different solid phases, such as collisions, aggregation or coalescence. This would amount to considering the density f in the phase-space $(\boldsymbol{x}, \boldsymbol{u}, St)$. An extra-term in the Liouville equation would then appear of the form $\partial_{St}\left[G(St)f\right]$. The kernel $G(St)$ stands for how a specific phase is affected by the medium or possibly by the other phases, like an evaporation rate for example. The variable St is used here for generality but the volume v of the particle is often considered as the additional mesoscopic variable, as in the Williams equation (Williams, 1958) dealing with the issue of polydisperse sprays. Actually, approaches have been proposed to integrate this equation in the limit of small inertia, i.e. when velocity dispersion is negligible. Those originated in Tambour (1980) with the idea of sectioning the v dimension. Later on, this method has been named Eulerian multi-fluid methods. Each fluid is represented as a statistical average of the distribution in each volume section. This method has received later on multiple improvements (Laurent *et al.*, 2004; Fox *et al.*, 2008). The dynamic in the volume space can also be performed with a finite-volume method, and the issue of numerical diffusion is also present. Fox *et al.* (2008) have presented a method called *direct quadrature method of moments* that seems to reduce this spurious diffusion. In those cases, it was stressed that the performance of such

Eulerian descriptions was still competitive compared to Lagrangian simulations (Laurent & Massot, 2001), thanks to easier parallelisation and easier treatment of fragmentation and coalescence. However, when the phase-space dimension has to be further increased to account for velocity dispersion, the issue of competition with Lagrangian methods is not that trivial.

Another extension of the proposed would be to use it for developing Large-Eddy Simulation of particle suspensions. Basically, LES for the fluid phase consists in filtering the Navier-Stokes equations. The same technique applied to the kinetic equation yields:

$$\partial_t \langle f \rangle + \nabla_x \cdot (\boldsymbol{v} \langle f \rangle) + \nabla_v \cdot \left(\frac{\langle \boldsymbol{u} \rangle - \boldsymbol{v}}{\tau_p} \langle f \rangle \right) = - \frac{\nabla_v \cdot (\langle \boldsymbol{u} f \rangle - \langle \boldsymbol{u} \rangle \langle f \rangle)}{\tau_p}, \tag{6.1}$$

where $\langle\rangle$ denotes a spatial averaging (RANS method), or filtering (LES). The right-hand side denotes the interaction between particles and sub-grid scale fluid eddies and is the main object that needs to be modelled. For instance, Zaichik *et al.* (2009) have expressed this right-hand side in terms of a Gaussian integration by parts, yielding additional diffusive terms, but other closure approached can easily be tested. Figure 4.14 showed that the lattice particle method was able to reproduce the correct spatial density even when using a resolution $N_x^2 = 512^2$ against a resolution of 2048^2 gridpoints for the fluid velocity field. This is equivalent to considering a filtered velocity field in the kinetic equation (4.14) and constitutes a promising result in the framework of LES. It would be worth comparing the method we have developed in this thesis with LES approaches with modelled closures.

6.3 Turbulence modulation by small heavy particles

In chapter 5, numerical simulations have been performed in order to explore the effect of two-way coupling in a system with heavy particles in two dimensional direct cascade. This study has been carried out in the two asymptotics of small and large Stokes numbers, using two different models for the particle transport. Chapter 5 reproduces the scientific article that presents the main results from this work in the case of the Large Stokes numbers. In this asymptotics, the numerical method presented in 4 has been adapted in order to include the back-reaction from the particles onto the fluid. The Eulerian formulation of the particle population in the position space allows for an easier and more natural implementation of two-way coupling. In addition, no closing, model or reconstruction for the particle to fluid interaction is needed contrary to Lagrangian-Eulerian methods, although much progress has been made in the last few years in this domain (see, for instance, Gualtieri *et al.* (2015); Ayala *et al.* (2007)).

Several statistical quantities have been measured for various coupling intensities, defined by the mass ratio ϕ_m. An effect directly observable when looking at the instantaneous density is the increase of particle clustering outside high-vorticity regions. The impact of two-way coupling on intermittency was also measured. In the two-dimensional direct enstrophy cascade, the distribution of vorticity gradients is characterised by broader tails than

Gaussian distributions. The action of the particles was shown to broaden this distribution, i.e. to increase its flatness, as the mass load ratio is increased.

Gravity has been neglected in the dynamics we considered. This is a valid approximation when fluid accelerations are stronger than g and when the fluid turbulent velocities are larger than the particle settling speed, which is more and more valid when the particles are not too massive. However, we presented an analysis with increasing mass load ϕ_m, i.e. with particles more and more massive with respect to the fluid. It would be interesting to include the gravity force to see how it affects the statistics we have presented. For example, experiments at large Stokes numbers carried by Hwang & Eaton (2006) showed that turbulent attenuation was uniform in the horizontal plane, in agreement with our measurements, and less pronounced in the vertical direction at high wavenumbers .

Simulations have been performed in the two-dimensional direct enstrophy cascade. In that case, a single time scale $\tau_f = (2Z)^{-1/2}$ can be defined (Z denotes here the average enstrophy of the fluid flow). However, in the inertial range of two-dimensional inverse energy cascade or three-dimensional direct cascade, the characteristic time of the flow $\tau_f = \epsilon^{-1/3} r^{2/3}$ depends on the observation scale r. So does the particles Stokes number with $St(r) = \tau_p/\tau_f \propto r^{-2/3}$ which is a decreasing function of r. For a given r^* in the inertial range, we may consider a time scale τ_p^* such that $St(r^*) = 1$. In this situation, the large scales $r > r^*$ would be affected by low-inertia particles the small scales $r < r^*$ by large-inertia particles. Following our results, the large scales would be destabilised by the Kelvin-Helmholtz instability at the large-sized-vortices boundaries and the energy spectrum less steep, while small scales would be damped with the energy slope.

Appendices

APPENDIX A

Software details

In these appendices, I discuss briefly about softwares that I have developed and used during my PhD thesis. The goal is to provide a transparent view to the reader about various numerical implementations and physical aspects.

A.1 GPU2DSOLVER

This C++-library was designed to solve Navier–Stokes equations in two dimensions in a periodic box.

A.1.1 Numerical scheme

The library solves the incompressible two dimensional Navier–Stokes equations for the stream-function $\psi(\boldsymbol{x}, \boldsymbol{y}, t)$ from which the flow velocity is derived by $\boldsymbol{u} = \boldsymbol{\nabla}^{\perp}\psi = (\partial_y\psi, -\partial_x\psi)$. This allows one to integrate only one quantity instead of the two velocity components, which saves memory while ensuring incompressibility. The equation solved is equation (2.18), and is discretised in time using second order Runge–Kutta scheme.

The numerical integration relies on the pseudo-spectral method, in which the solution is decomposed onto a Fourier basis, and the evolution equation is solved for the Fourier coefficients $\hat{\boldsymbol{u}}(\boldsymbol{k})$, with $\boldsymbol{k}$ the wave-number of the mode.

The general advantage of the spectral methods (Orszag, 1969), especially involving Fourier decompositions suited for periodic boundary conditions, stands mainly in their exponential rate of convergence for smooth fields (faster than any polynomial in the grid size) (Bardos & Tadmor, 2015; Canuto *et al.*, 2012). Furthermore, the evaluation of the derivative is much more precise than finite difference methods. Indeed, one does not need to approximate derivatives using a finite number of points because it is straightforwardly given by the multiplication with the wave-number vector in the Fourier space. For instance: $\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = \int e^{i\boldsymbol{k}\cdot\boldsymbol{x}}\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k})$.

Another advantage lies in the existence of fast algorithms to compute Fourier transformations which only involve $N \log N$ operations,where N is the number of field elements, instead of N^2 as would perform a naive implementation (Walker, 1996; Brigham & Brigham, 1974). In addition, those algorithms were also been adapted and optimised for various parallel architectures (for memory distributed cluster, general purpose graphical processing units, many-integrated core chips...). Spectral methods are also less expensive than finite element methods for simple open geometries.

The term pseudo means that a part of the integration is done in the physical space. Indeed, fields products appearing in the Navier-Stokes equation (in the non-linear, convective term) translate into convolution in Fourier space involving N^2 operations. Transforming those fields in the physical space to preform this element-wise product and transforming them back again in the Fourier space only involves $4N\log(N) + N$ operations, instead of N^2 for the convolution. The field transformed back into Fourier space must be de-aliased: the nonlinear term computation creates non-physical high-k modes. The classical 2/3 zero-padding rule is used, setting $\psi(\boldsymbol{k}) = \psi(\boldsymbol{k})\Theta(k_{\max} - \|\boldsymbol{k}\|)$, i.e, $\psi(\boldsymbol{k}) = 0 \;\forall\, \boldsymbol{k}$ s.a. $\|\boldsymbol{k}\| > k_{\max}$, with $k_{\max} = N/3$. In other words, all modes outside a circle of radius $k_{\max}$ are removed.

The potential ψ is initialized to $\psi(\boldsymbol{x}) = 0$ by default or may be read from an external file. This allows simulations to restart from a previous state, for example equilibrated flows in high resolutions. It may also be prescribed analytically to test for special solutions or to generate benchmark flows, like cellular ones (see section 4.5.1).

Viscous dissipation and hyper-viscosity

The integration of the dissipative term is made implicit:

$$\psi^{t+\Delta t}(1 + \nu \|\boldsymbol{k}\|^2) = \psi^t. \tag{A.1}$$

Hyper-viscosity is also implemented, for which viscous dissipation operator uses a higher power of the Laplacian: $\hat{D} = -\nu\nabla^{2q}$, yielding in Fourier space $\hat{D} = -\nu(-1)^{q+1}k^{2q}$. This is often used to extend the inertial scaling range in the energy and enstrophy spectra and is especially suited when studying dynamics in this regime, such as the Richardson pair separation issue in the inverse energy cascade (see chapter 3). In this formulation for $\hat{D}$, ν has to be scaled accordingly and may go down to 10^{-47} for $q = 16$ (Smithr & Yakhot, 1994). To avoid having to renormalise the viscosity and use a value of the similar order for different resolutions, we used for the operator the form $\hat{D} = -\nu k^2 \left(\frac{k}{k_{\max}}\right)^{2(q-1)}$.

A.1.2 Forcing

Conservation equations for the fluid momentum writes:

$$\partial_t(\rho_f \boldsymbol{u}) + \boldsymbol{u}\boldsymbol{\nabla} \cdot (\rho_f \boldsymbol{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\Sigma} + \boldsymbol{f}. \tag{A.2}$$

These equations are dissipative in the case of $\boldsymbol{f} = \boldsymbol{0}$ due to the viscous forces in the constrain tensor Σ (see section 2.1), which physically result from small-scale friction between fluid particles. Starting from a non zero initial condition, energy would thus decrease until the flow is at rest ($\boldsymbol{u} = \boldsymbol{0}$).

Meanwhile, turbulence may arise due to non-linear transfer of energy through the wave-numbers (see 2.2.2). Studies of such non sustained systems in various situations, is called *decaying turbulence* in the literature. However, all experiments that I present in this manuscript involve flows in statistically steady state, hence which are sustained by an external source of energy. In real situations, such a forcing may come from mechanical agitation (propellers, high speed flows through a pipe...), or thermal effects (nuclear and chemical reactions in flows...).

The numerical forcing that is implemented in GPU2DSOLVER is volumetric: momentum is added in Fourier space to modes in a given range of wave-numbers $[k_{\text{inf}}, k_{\text{sup}}]$. For example, a forcing at large scales $|k| \ll$ is often associated to *stirring*.

Forcing methods is still a matter of debate depending on the applications. Indeed, although it is needed to reach a stationary steady state, one has to make sure that the forcing will not influence the statistics of the flow. For example, it is generally admitted that small scale dynamics are decoupled from large scale ones for sufficiently large Reynolds numbers (Eswaran & Pope, 1988).

One can use multiple numerical methods (this list is not exhaustive):

- Freeze the forced modes to a constant value, fixing the large scale structures.
- Add a constant quantity to the forced modes: $f = \text{cst}$.
- Renormalise the total energy in the forced region so that at every time step such that:
$$\int_{k_{first}}^{k_{end}} |\psi|^2(\boldsymbol{k}) d\boldsymbol{k} = \text{cst}. \tag{A.3}$$
- Use a stochastic process for f. This one is widely used due to its randomness which is believed to be more realistic.

The forcing methods cited above have all been implemented in GPU2DSOLVER. In practice, only the last one, non deterministic, was retained in the studies I have carried, mainly for the reason of being more physically realistic. Two random processes are implemented: for the first one, the force follows an Ornstein-Uhlenbeck process:

$$\mathrm{d}f(\boldsymbol{k}) = -\theta(f(\boldsymbol{k}) - \mu)\mathrm{d}t + \sigma \mathrm{d}W(\boldsymbol{k}). \tag{A.4}$$

It consists of one drifting term, $\theta\mu\mathrm{d}t$, one damping term, $-\theta f(\boldsymbol{k})$, and one diffusion term, Gaussian, of variance σ^2. The processes are independant, i.e $\langle \mathrm{d}W(\boldsymbol{k}_1)\mathrm{d}W(\boldsymbol{k}_2)\rangle = \delta(\boldsymbol{k}_1 - \boldsymbol{k}_2)$. These processes have a non zero correlation time given by $\sigma^2/2\theta$. μ is the asymptotic average value and is always set to 0 in the studies I have carried.

$$f(\boldsymbol{k}, t) = f(\boldsymbol{k}, t - \Delta t) \exp(-\theta\Delta t) + \sqrt{\sigma\Delta t}\, \xi(\boldsymbol{k}, t) \tag{A.5}$$

with ξ a number from the normal distribution $\mathcal{N}(0, 1)$.

For more details about this process numerical integration, see Gillespie (1996); Honeycutt (1992*a*,*b*).

The other stochastic forcing implemented is a white noise:

$$f(\boldsymbol{k}, t) = \sigma\xi(\boldsymbol{k}, t) \tag{A.6}$$

This kind of forcing is delta-correlated in time: $\langle f(t)f(t')\rangle = \delta(t - t')$ and is especially suited for small scale forcing, as the correlation time of the velocity decreases with the scale.

In practice, for both forcing types, σ is normalized in the following way:

$$\sigma' = \sigma/(k^2\, 2\pi(k_{sup} - k_{inf})\, (k_{sup} + k_{inf})/2). \tag{A.7}$$

The k^2 factor removes the k^2 dependence arising by the growing number of modes in shells between k and $k + 1$. $2\pi(k_{sup} - k_{inf})$ accounts for the number of forced modes, and the $(k_{sup} + k_{inf})/2$ normalises by the middle k_f. This normalisation allows one to use a forcing amplitude independent of the spectral forcing band width, the mean forced mode and the increase of the number of modes in each shell $[k, k + 1]$.

Simple Euler temporal discretisation is used:

$$\psi(\boldsymbol{k}, t) = \psi(\boldsymbol{k}, t - \Delta t) + f(\boldsymbol{k}, t). \tag{A.8}$$

This is sufficient because we are not interested in a precise resolution of the forcing processes, as they are just random sequences.

A.1.3 Parallelisation

The software makes exclusive use of general-purpose graphical programming units (GPGPU) to integrate the flow, using the proprietary CUDA C-interface (Nickolls *et al.*, 2008). Indeed, this architecture was chosen because it is at present the best suited for massively independent calculations, like in our case the update of all fluid Fourier modes, or linear algebra operations. Furthermore, fast Fourier transforms also benefit from very well optimized algorithms for GPU architectures. In my case, use was made of the cuFFT library. The cuRAND library was also used to generate pseudo-random sequences for the forcing. One sequence was generated for each forced mode in a way that allows one to simulate reproducible flows. A quick benchmark with a pseudo-spectral 2D solver showed that simulations using one single graphic card perform as well as distributed memory simulations (using message passing interface) with ~ 120 cores. Nevertheless, the choice of this proprietary language implies that the software is only capable to run on systems with CUDA-capable graphic cards.

It should be stressed that the very limiting factor for high resolution fluid simulation is the available memory. The GPUs I have used during my thesis were middle-range Tesla M2050 graphic cards from Nvidia which had a video memory of $\sim 3.5GB$. I was able to perform fluid computations with resolutions up to $N_x^2 = 8192^2$, which only corresponds to $N_x^3 \sim 400^3$ in three dimensions while state of the art three dimensional computations go up to $N_x^3 = 8192^3$ especially in cosmological simulations (Yeung *et al.*, 2015). Newer generations of graphical processing units can go up to $O(10GB)$, but it is still limited to perform high resolution three dimensional simulations on a single graphic card. It is thus necessary to couple those acceleration devices with distributed memory architectures using message passing protocols among many computational nodes. The performance then hinders from inter-node communications. Nowadays high performance computing codes show to linearly scale up to $\mathcal{O}(10^5)$ processors for $\mathcal{O}(6144^3)$ resolutions (see, for instance, Mininni *et al.* (2011)).

Figure A.1 displays a benchmark to assess the performance of the software. Number of points integrating by a full Runge-Kutta 2 step per second is shown as a function of the resolution N_x. It corresponds to the the number of evolution of the full 2D field multiplied by N_x^2. This quantity was computed by averaging over a fixed number of 600 time steps.

For a single CPU, it would behave as N_x^0, i.e. constant. In MPI implementations, benchmarks are often presented as the time (in seconds or minutes) per time step as a function of the number of processor for a fixed resolution. Performance generally scales linearly with the number of processor until it eventually reaches a plateau, displaying the bottleneck of messages communication which limits the performance. The metric displayed in Figure A.1 is rather analogous, the constant behaviour being the proof of the absence of an additional limiting bottleneck. Notice that in this case, as a single GPU card is used in this case, the number of available threads and cores is fixed.

A slope greater than 0 indicates that the scaling is better than a single-threaded execution, and this is what is indeed observed for small N_x. This can be explained by an increase in the device occupancy: increasing the resolution, hence the number of points to be treated, exposes more parallelism, i.e. there are less idle threads and the GPU card is more efficiently exploited.

The saw-tooth like behaviour is due to the alternating resolution between N_x being or

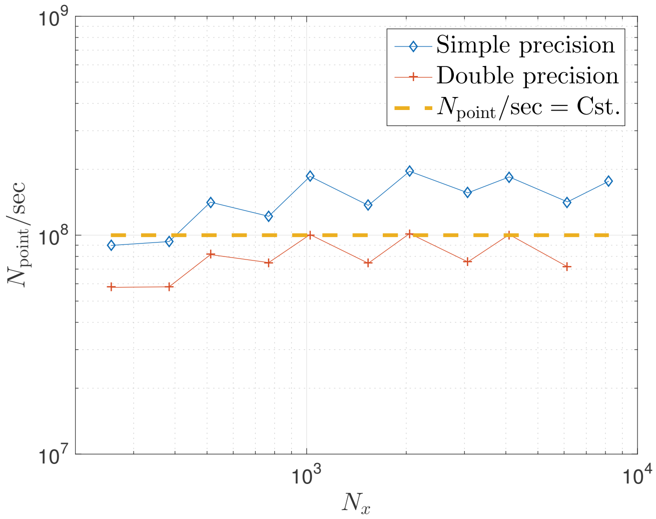


Figure A.1: Benchmark of the GPU2DSOLVER software performed on a middle-range GPGPU, the Nvidia Tesla M2050. The number of points integrated per second is represented as a function of the resolution N_x up to $N_x = 8192$.

not a pure power of 2. FFTs algorithms perform better for combinations of prime number powers, i.e. $2^a 3^b 5^c 7^d ...$, the lower the number the better. In this case, N_x is alternatively 2^i and $2^{i-1}3$, with $i = 8, ..., 13$, excepting $2^{12}3$.

This code is under CECILL-C license and available on demand.

A.2 LAGSRC2D

This C++ library was designed to simulate emission of particles from a punctual source and removal from the domain following a custom criterion. The particle system may then be prescribed an external fluid velocity for its transport. The work in which it was used can be found in chapter 3.

A.2.1 Numerical implementation

A class is built to represent the whole particle system. Because it is generally more optimal to pack in memory contiguous data, mainly to facilitate cache reuse (Cheng *et al.*, 2014), it was decided to encompass in it the arrays representing particle properties, i.e. positions, velocities, accelerations, unique identifiers and ages. Each property is a GPU array, more precisely Thrust device vectors, from the Thrust library distributed with the CUDA toolkit (Bell & Hoberock, 2011). This library offers the advantages of transparent arrays utilisation and C++ standard template library (STL)-style high level algorithms.

Another way of designing the system would be to use fixed-length arrays, with a size large enough to contain the desired maximum number of particles present in the system. Then when particles exit the system due to some boundary conditions, they are reinjected. This solution allows a gain in performance, because less time is spent to reorganize memory by removing and adding elements at random places in the vectors. However, it would create spurious temporal correlations between the injection intensity and boundary conditions. Regarding the physical issue for which the library was designed, this is an undesired effect.

The design of this class was part of the motivation to work on the SoAx library project (see appendix B). Indeed, this last library allows one to manipulate a particle system and, through macro-generated function, to add very simply properties against which class methods can be called.

LAGSRC2D is able to handle two types of removal conditions: particles are removed after an age $A_{\max}$ or after a maximum distance from the injecting source. At a frequency chosen by the user, the system checks for the exit condition and erases elements for which this condition is true. This is done through a call to C++ standard library functions remove_if and erase, which gives a linear complexity of N comparisons and some n deletions, with generally $n \ll N$.

The dynamical update of particles position and velocities make use of the second order Adams-Bashforth scheme (Jeffreys & Jeffreys, 1999):

$$v_i^{t+\Delta t} = v_i^t + \Delta t(\frac{3}{2}A_i^t - \frac{1}{2}A^{t-\Delta t}) + \sqrt{2\Delta t \sigma}\xi(t). \tag{A.9}$$

Δt is the time step, A_i^t stands for the $i-$component of the acceleration at time t, ξ a random number from $\mathcal{N}(0,1)$ distribution and σ the noise amplitude parameter to add a random perturbation. This perturbation typically results from random shocks with fluid particles. In chapter 3, $\sigma = 0$.

When the knowledge of the flow velocity at the position of the particle $\boldsymbol{u}(\boldsymbol{X}_p)$ is required, its value is determined by a bicubic interpolation, thus involving 16 points around the particle.

Files for chosen properties are written on the disk with the time of the output and a restart is then possible from any of those output by specifying the restart time.

A.2.2 Injection rate

A parameter N_I*, called *injection intensity* is used to control the number of particles emitted in a duration of $t = 1$. Another parameter determines how often the injection takes place, the *injection period* T. The number of particles to be emitted during each period is then $N_{I_T}^* = N_I^*/T$, which may be a floating point number.

Then, to ensure that an integer number of particles is injected per period, $N_{I_T}^*$ is adjusted (simply rounded) so that the computed N_I may be a bit higher or lower than requested. This number may then be zero, and if so the user is issued a warning and has to modify the injection period. The maximum deviation expected between required N_I^* and actually computed N_I is thus $|N_I - N_I^*|/N_I^* = \frac{1}{2}\frac{t/T}{N_I*}$. In practice, injection is done at the beginning of each period.

This code is under CECILL-C license and available on demand.

A.3 LOCA: Lattice model for heavy particles

This C++ library was designed to simulate transport of heavy particles in externally-prescribed flows. The challenge is to take into account the particles velocity dispersion at a given point in space. For more details on this physical issue, see chapter 4.

As explained in chapter 4, the equation (4.14), which is a transport equation in the position-velocity-phase-space for a scalar quantity, is to be discretised and represented in 4 dimensions.

A.3.1 Finite volume fluxes

Inertial particles heavier than the fluid cluster in regions of low fluid vorticity. Initial uniform distributions will thus develop strong density fronts in the position space (see for example section 4.5.2), and also in the two dimensional velocity space, where a finite volume scheme is used for the numerical integration. Indeed, particles tend to relax toward the fluid local velocity $\boldsymbol{u}(\boldsymbol{x})$ with a characteristic time τ_p. The distribution width around this velocity is thus an increasing function of τ_p, and when $\tau_p \to 0$, particles perfectly align their velocity to the fluid one, corresponding to a δ-function in the velocity space.

The finite volume should then be able to adapt to strong gradients when necessary. Furthermore, positivity preservation is a necessary condition when dealing with positive quantities such as mass, or substance concentration, as it is often the case in engineering simulations. This property can be ensured requiring the TVD (total variation diminishing) property of the scheme, which ensures not to create new extrema in the function. I thus needed a finite volume discretisation which gave both good accuracy, that could automatically adapt to strong fronts, and that preserved positivity.

The natural solution is the use of *flux limiters*, which are function of the local gradients, either flux or state gradients. They are used to tune the part of the flux that comes from a high order scheme and the one that comes from a low order scheme. A lot of limiters exist, and after some trials among a few other TVD ones, the one from Koren (1993) was chosen. It offered the best accuracy, although only the test for one spatial resolution in the one dimensional case ($N_x = 2048$) was performed. Its expression writes $K(r) = \max[0, \min(2r, (2+r)/3, 2)]$, where r is the local gradient. In the case of a positive net force, $F > 0$ at the interface between cell i and $i+1$ (the *upwind direction*),

$$r = \frac{f_i - f_{i-1}}{f_{i-1} - f_{i-2}}. \tag{A.10}$$

Flux at this cell interface is then computed by (Hundsdorfer *et al.*, 1995):

$$\phi_{i+\frac{1}{2}} = f_i F_i + \frac{1}{2} K(r)(F_i f_i - F_{i-1} f_{i-1}). \tag{A.11}$$

The density field is updated via a Euler temporal discretisation:

$$f_i^{t+\Delta t,*} = f_i^t + \frac{\Delta t}{\Delta v}(\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}}). \tag{A.12}$$

When including diffusion, the fluxes at the cell interface are applied subsequently:

$$\phi^D_{i+\frac{1}{2}} = \frac{\left(f_{i+1}^{t+\Delta t,*} - f_i^{t+\Delta t,*}\right)}{\Delta v}, \tag{A.13}$$

$$f_i^{t+\Delta t} = f_i^{t+\Delta t,*} + \frac{\kappa \Delta t}{\Delta v}\left(\phi^D_{i+\frac{1}{2}} - \phi^D_{i-\frac{1}{2}}\right). \tag{A.14}$$

A.3.2 Dynamic grid resizing (DGR)

The domain grid in the velocity space consists of regular square cells disposed in the interval $[-V_{\max}, +V_{\max}]^d$ with d the dimension. Fixing this parameter forces the user to know *a priori* the value of the particles rms-velocity needed to be resolved. It can be estimated in some cases, as it was in the study discussed in chapter 4. In this case, some pre-run with Lagrangian particles were carried to estimate their rms-velocity, and this property was known to decrease with $St^{1/2}$ (St is the Stokes number of the particles). Of course this knowledge is not always guaranteed. In particular, this is a problem faced for the study described at chapter 5. In that case, the flow exerts drags on the particles and the particles react back on the fluid. The system is thus coupled and it was a quantity of interest to determine how this velocity varied with the intensity of the coupling (defined via the mass ratio between the particle population and the fluid total mass). A tendency which could also depend on the particles inertia. Furthermore, the importance to provide a good value for the parameter $V_{\max}$ is shown in section 4.4.2, where its impact on the error on the density field was assessed.

It was thus decided to implement a dynamic re-gridding to be able to dynamically adjust the parameter $V_{\max}$ depending on the particle dynamics.

It is important to note that this method is different from *adaptative mesh refinement*, or *AMR*. This last technique is used in the scientific community for methods generating more densely gridded parts of the domain. It is widely used for large scale computing, for example in astrophysics (Teyssier, 2002; Mignone *et al.*, 2011; Bryan *et al.*, 2014), using a tree-based decomposition, advantageous for systems where small regions need large refinement.

In our case, the number of cells is constant, and the $2d$ grid remains uniform. Only the parameter $V_{\max}$ is adapted, so that the dynamic of the particles is better resolved.

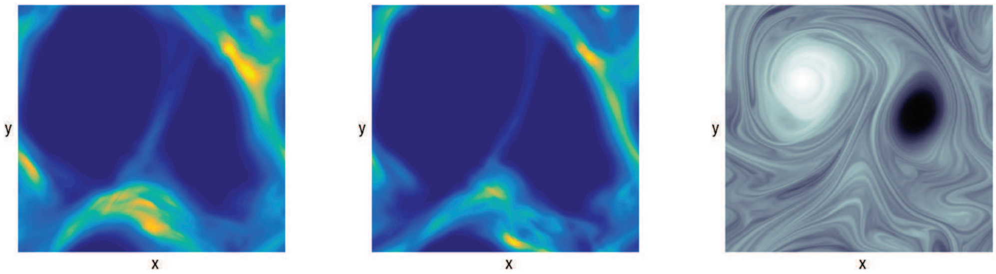


Figure A.2: Particle density field without (left) and with (center) DGR. Left simulation uses a fixed value of $V_{\max} = 4$ while the real value is ~ 3. Right simulation starts with $V_{\max} = 4$, resizing the velocity grid every $10\tau_p$ at $V_{\max}/v_{p_{\text{rms}}} = 2$. It resolves better the mass voids due to high vorticity regions (corresponding vorticity field shown on the right). Resolution is $N_x^2 \times N_v^2 = 1024^2 \times 13^2$ and $St \sim 1$.

This adjustment takes place at a frequency specified by the user, and the desired ratio $V_{\max}/v_{p_{\text{rms}}}$ is also prescribed. At each adjustment, the resolution in the velocity space $\Delta v = \frac{2V_{\max}}{N_v}$ is modified and so is the time step $\Delta t = \Delta x/\Delta v$ which is a necessary criterion for agreement with the discrete displacements on a lattice (see section 4). The transformation between the particle population before DGR, f^{old}, and after DGR, f^{new} is made through a bilinear interpolation in the velocity space. If $V_{\max}^{\text{new}} < V_{\max}^{\text{old}}$, then all the mass $\int_{\Omega_x} \int_{V_{\max}^{\text{new}}}^{V_{\max}^{\text{old}}} f^{\text{old}}(v)\ \mathrm{d}^d x \mathrm{d}^d v$ is discarded (Ω_x is the spatial domain), then the remaining field is rescaled to conserve the total mass. For example, assuming a Gaussian distribution for the velocities, if $V_{\max}/v_{p_{\text{rms}}} = 2$, then maximum $\sim 4.5\%$ of the mass will be impacted.

Of course, one has to provide a minimum $V_{\max}$. Otherwise, Δv would tend to 0 and the stability criterion would not be fulfilled any more (see section 4.3 for more details on stability). This is particularly the case when starting from rest initial condition.

This technique was tested on a benchmark with a two dimensional direct enstrophy cascade at resolution $N_x^2 \times N_v^2 = 1024^2 \times 13^2$. Two simulations are performed using the same stochastic forcing, with and without DGR. Figure A.2 displays two instantaneous density fields for $St \sim 1$. One can see that DGR allows to reduce numerical diffusion and to better resolved the clustered trajectories outside the fluid vortices. Systematic study of the impact of this grid resizing would require to assess the numerical error between Lagrangian particles and corresponding reconstructed Eulerian mass fields, as a function of the ratio $V_{\max}/v_{p_{\text{rms}}} = 2$ and the frequency of adjustment.

A.3.3 Parallelisation

Although the flow is computed using GPU parallelisation, the lattice method was integrated on the CPU using shared memory parallelisation with the OPEN-MP library. The overhead of fetching the fluid data located on the GPU was compensated by the fact that the time step of the flow was always smaller than the time step of the lattice particles, allowing for a not so frequent memory fetch. It was indeed verified through profiling that the part of the computation time dedicated to the memory transfers between CPU and GPU was negligible ($\mathcal{O}(1\%)$). Furthermore, this allows to release pressure on the scarcely available memory on the GPU, and to perform CPU computations in parallel to the GPU.

Finally, the algorithm was performing better using shared memory than GPU acceleration. This is easily interpretable: GPU acceleration functions better when processing adjacent memory locations. Because the full particle distribution f is stored in a linear array in memory with the velocity being the fastest varying dimension, the GPU parallelisation is efficient during the acceleration step, but certainly not during the streaming step, where lots of memory movements have to be performed by lightweight threads. Actually, this issue is common with the Lattice-Boltzmann GPU acceleration, for which efficient algorithms have been proposed (see, for instance, Habich *et al.* (2011)).

This code is under CECILL license and available on demand.

APPENDIX B

SoAx: a convenient and efficient C++ library to handle simulation of heterogeneous particles in parallel architectures

A large variety of natural and industrial applications require scientists and engineers to use a collection of a large number of objects encompassing multiple attributes. For example, in the research framework of this thesis, these objects are solid particles suspended in turbulent flows and the attributes are, or can be, the positions, velocities, mass, electric charge, concentration, lifetime, etc. When designing a code to handle these attributes, it quickly appears that adding more properties becomes cumbersome in terms of code maintenance and reusability. Indeed, for performance reasons, it is desirable that each attribute be declared as a separate, linear array in memory, promoting cache reuse. The trace of this array has then to be kept and propagated in all the members of the class representing the particles. Adding a new attribute forces then one to pass in review all the methods of the class encompassing these arrays.

The SoAx library basically solves this problem. It is placed under GPLv3.0 license.

Although the *chained-list* model may be more advantageous in some situations, it is not when repeating the same operation for each attribute on every particles and repeatedly a large number of times as in numerical integrations of differential equations. This is called *number crunching*.

A list of its capabilities are:

- Macro generated class members. Each attribute may be given a handy name, and each additional one is automatically added to member functions.
- GPGPU support. Users may choose that their data reside on a GPU device. In that case, all commands and computations are executed on the device. Transfer between

CPU and GPU only occurs when the user explicitly retrieves the data from the GPU.

- MIC support.
- Expression templates: naive implementations of chained operations (for example: a = (b+c)*d/(e-f)) between arrays of data results in multiple traversal of the data, operations between executed between pairs of arrays, thus leading to poor cache utilisation. Expression templates, that arose since the great possibilities offered by template metaprogramming in C++, allow to embed the succession of operations to be performed an to apply them in a single traversal of the data (see, for instance, Iglberger *et al.* (2012)).
- MPI-ready. Particles may be individually retrieved, shared among processors and saved to .

Performances of SoAx are measured as a function of the number particles on several architectures (MIC, GPU). For each case, they are compared to a classical implementation, i.e. not using macro generated class members or expression templates, to assess for the validity of the implementation.

The paper issued from this development submitted to *Computer physics communications* is reproduced hereafter.

SoAx: A generic C++ Structure of Arrays for handling particles in HPC codes

Holger Homann*, Francois Laenen

Laboratoire J.-L. Lagrange, Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, F-06304 Nice, France

---

Abstract

The numerical study of physical problems often require integrating the dynamics of a large number of particles evolving according to a given set of equations. Particles are characterized by the information they are carrying such as an identity, a position other. There are generally speaking two different possibilities for handling particles in high performance computing (HPC) codes. The concept of an *Array of Structures* (AoS) is in the spirit of the object-oriented programming (OOP) paradigm in that the particle information is implemented as a structure. Here, an object (realization of the structure) represents one particle and a set of many particles is stored in an array. In contrast, using the concept of a *Structure of Arrays* (SoA), a single structure holds several arrays each representing one property (such as the identity) of the whole set of particles.

The AoS approach is often implemented in HPC codes due to its handiness and flexibility. For a class of problems, however, it is know that the performance of SoA is much better than that of AoS. We confirm this observation for our particle problem. Using a benchmark we show that on modern Intel Xeon processors the SoA implementation is typically several times faster than the AoS one. On Intel's MIC co-processors the performance gap even attains a factor of ten. The same is true for GPU computing, using both computational and multi-purpose GPUs.

Combining performance and handiness, we present the library SoAx that has optimal performance (on CPUs, MICs, and GPUs) while providing the same handiness as AoS. For this, SoAx uses modern C++ design techniques such template meta programming that allows to automatically generate code for user defined heterogeneous data structures.

Keywords: keyword1; keyword2; keyword3; etc.

PROGRAM SUMMARY/NEW VERSION PROGRAM SUMMARY

Manuscript Title:SoAx: A generic C++ Structure of Arrays for handling particles in HPC codes
Authors: Holger Homann, Francois Laenen
Program Title: SoAx
Journal Reference:
Catalogue identifier:
Licensing provisions:
Programming language:C++
Computer: CPUs, GPUs
Operating system: Linux
RAM: bytes
Number of processors used:
Supplementary material:
Keywords: Keyword one, Keyword two, Keyword three, etc.
Classification:
External routines/libraries:
Subprograms used:
*Catalogue identifier of previous version:**
*Journal reference of previous version:**
*Does the new version supersede the previous version?:**
Nature of problem:

Solution method:

*Reasons for the new version:**

*Summary of revisions:**

Restrictions:

Unusual features:

Additional comments:

Running time:

References

[1] Reference 1
[2] Reference 2
[3] Reference 3

* Items marked with an asterisk are only required for new versions of programs previously published in the CPC Program Library.

1. Introduction

Particles are at the heart of many astrophysical, environmental or industrial problems ranging from the dynamics of galaxies over sandstorms to combustion in diesel engines. Investi-

---

*Corresponding author.
E-mail address: holger.homann@oca.eu

gating such problems require generally integrating the dynamics of a large number of particles evolving according to a given physical laws. Examples are N-body simulations in cosmology [1], particle in cell codes (PIC) exploring plasma physics [2] or hydrodynamic simulations studying Lagrangian turbulence problems [3]. Such kind of numerical simulations have in common that that they are numerically expensive meaning that they rely on number crunching, i.e. an enormous number of floating point operations. Studying the particle dynamics during a finite time interval requires the numerical integration of the underlying equations of motion over many time steps so that the particle data (position, velocity, ...) is used in simple but numerous repeated operations. The performance of such operations depend in a crucial way on how the particle data is stored and accessed.

Modern supercomputers are often indispensable for studying challenging problems. Their architecture got more and more complex in recent years. The today's fastest computers consists of several performance sensible components such as multi-level caches, vector units based on the 'single-instruction multiple-data' (SIMD) concept, multi-core processors, many-core (MIC) and GPU accelerators. Evidently it is important to make use of all these components to optimize the performance of a numerical code.

Particles can carry different properties such as an identity, a position or a mass. In programming languages such as Fortran, C or C++, the data types `int`, `double` and `float`, could be chosen to represent the former particle properties. (In this paper codelets (serving as implementation examples) will always be given in C++, but the reasoning will be kept general so that it will similarly apply to Fortran and C.)

In C++, particles can be implemented as a heterogeneous structure

Listing 1: Particle structure storing the data of one particle

```
  struct Particle {
    int id;
    double position;
    float mass;
  }
```

This way, individual particles can easily be generated as objects (`Particle` p;) and modified (`p.id = 42`). A set of particles is then often handled by an array- or list-like structures (`std::list<Particle> pList;`) providing functionalities such as access, adding and removal of particles. Such an organization is called *array of structures* (AoS) as the particles are represented by a structure that is hold by an array (or list). Treating particles as objects is also convenient for transferring them from one process to another via the message passing interface (MPI) in parallel applications.

Another implementation strategy for handling a set of many particles is to use one structure that holds several arrays; one array for each particle property:

Listing 2: Structure of array containing one array per particle property

```
class PartArr
{
public:
  int* id;
  double* position;
  float* mass;
}
```

It is then convenient to add member functions to this class that perform operations on all the properties such as allocating memory:

Listing 3: Member function to allocate memory for particle property arrays

```
void PartArr::allocate(int n)
{
  number = new int[n];
  position = new double[n];
  mass = new float[n];
}
```

In the same way, member functions for adding, removing and other functionalities could be added. This kind of implementation is called *structure of arrays* (SoA) from the fact that in this case one structure handles a set of particles whose properties are represented by different arrays. `PartArr pArr;` creates a set of particles and an individual particle is referenced by the array index (`pArr.position[42]` returns the position of particle 42). A priory, particles cannot be extracted as individual objects from the structure `PartArr`. For this, a structure `Particle` (see codelet above) would be needed together with a function copying the array data for one index to the `Particle` member variables. From these considerations it is clear that AoSs are easier to implement and to use than SoAs.

AoS are also more extendable than AoS. Imagine one would like to reuse the above outlined particle storage implementation for a slightly different particle type that requires the addition of a property such as a charge. AoS are more flexible than SoA for this task: the novel property could be added to the `Particle` structure by simply adding the member variable `float charge;`. In the case of a SoA an array (through `float* charge;`) could be added to `PartArr`. But in turn, all member functions such as `allocate` would also have to be updated in order to treat the added array.

AoS seem to be the better candidate to store particles than SoA. However, SoA are faster in many circumstances (especially on MIC and GPUs) [4, 5] than AoS and we show that this is also the case for typical manipulations (such as trajectory integration) on particle data. By means of a benchmark modeling floating-point operations used in real codes we show that SoAs are typically several times faster than AoSs and that the performance of an AoS depends on the size (in terms of bytes) of the structure (`Particle` in the example above). In order to cope with the seemingly contradicting properties handiness, flexibility and performance, we present a generic implementation of a SoA called SoAx that has optimal performance while providing the same handiness and flexibility of an AoS.

This paper is organized as follows. In section 2 we benchmark the performance of AoS and SoA on CPUs. In section 3 we discuss a similar benchmark on GPUs and MICs. The generic C++ implementation of SoAx is presented in section 4. Conclusions are drawn in 5.

2. Benchmarking AoS and SoA on CPUs

In order to compare the performance of SoA and AoS we measure the execution time of a benchmark computation. The latter consists in performing an Euler advection time step for the position $\boldsymbol{x}$ of a set of particles

$$\boldsymbol{x} += dt\,\boldsymbol{v}, \tag{1}$$

dt denoting the time step (a floating point number) and $\boldsymbol{v}$ the velocity of the particle. This equation is a simple prototype for typical operations appearing in numerical codes. It consists, for each component, of two loads from the heap memory plus one for the constant dt, usually from the stack, and two stores in heap memory.

For benchmarking AoS we use the structure

Listing 4: Particle structure used in the benchmark. SIZE is the number of supplementary floats.

```
template<int SIZE>
class Particle
{
  public:
  float x[3];
  float v[3];
  float temp[SIZE];
};
```

where temp is a place holder for additional particle properties that might be necessary for the physical problem under consideration (such as a mass, an electric charge...) or the numerical algorithm (such as temporary positions and velocities for a Runge-Kutta scheme). In the case of SoA we simply use three heap-allocated C++ arrays for $\boldsymbol{x}$ and $\boldsymbol{v}$, respectively.

Typically, in numerical simulations many successive time steps are performed in order to integrate the particle dynamics. In our benchmark we therefore loop many times over the numerical implementation of (1). We use standard compilers with enabled optimization.

Figure 1 compares the normalized execution time for the SoA and AoS as a function of the particle number. The SoA is much faster than the AoS. Their relative performance is shown in Fig. 2. The SoA implementation is up to 25 times faster than the AoS and one gains at least a factor of two to three by using a SoA instead of an AoS.

The measured performance depends on the number of particles which is a consequence of the different cache levels of modern CPUs. Usually they provide three levels with sizes of 32 kByte (L1), 256 kByte (L2), and 8-40 MByte (L3). The colored arrows in Fig. 1 show the cache limits in terms of a the number of particles of a certain size (in terms of bytes). One observes that the performance is maximal when the L1 cache is filled and all particle data still fits into the L2 cache. When the particle data size exceeds the L2 cache, the execution time slightly increases. An important performance drop happens when data becomes larger than the L3 level. At that point data has to be transferred from the main memory that has a significantly lower bandwidth than the caches.

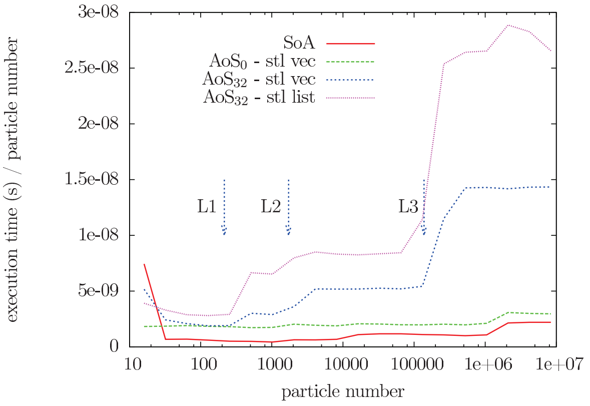


Figure 1: Benchmark comparing the performance of a structure of array (SoA) and an array of structure (AoS). The index SIZE in AoS$_{\text{SIZE}}$ denotes the number of supplementary floats in the structure Particle (List. 4).

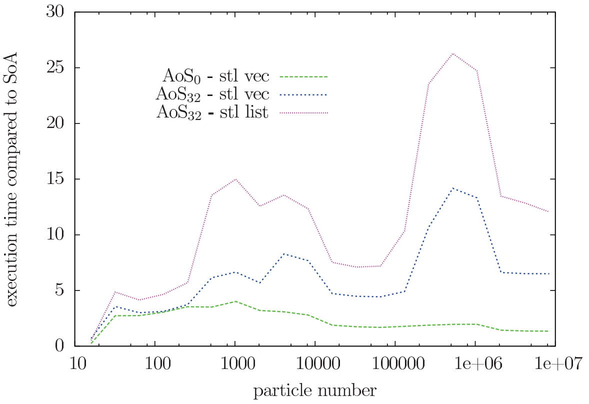


Figure 2: Execution time of an array of structure (AoS) relative to that of a structure of array (SoA). The index SIZE in AoS$_{\text{SIZE}}$ denotes the number of supplementary floats in the structure Particle (List. 4).

An important drawback of AoS is that its performance depends on the size of the particle structure. The more data (properties) this structure holds, that is to say the bigger it is, the more it fills the cache that in turn hinders performance. A particle with 32 additional floating point member variables (SIZE=32 List. 4) in is much slower than its slim counterpart. This problem is of course absent for SoA as all arrays are allocated individually and continuously in memory. Data (particle properties) that is not used in the execution loop will not be loaded into the cache.

The execution time of the AoS also depends on the container used to store the particle objects. A stl vector is significantly faster than than a stl list. We measure roughly a factor of two. This difference is due to the additional indirections involved for linked lists (such as the stl list). On the other hand, a list is faster in removing particles than a vector as the latter copies successive elements to keep the data continuous in memory. This drawback can be overcome when the ordering of particle is not important. In that case, a particle can be removed by simply overwriting it with the last particle. This strategy is used by default by SoAx.

The performance measured with a given benchmark naturally depends on the architecture of the CPU. However, it is important to note that the just discussed relative performance (SoA vs. AoS) will not or only weakly depend on the clock speed. But other differences, especially the vectorization units are important as we will show now. We will consider two different CPU architectures distinguished by the date of their commercial release. This sheds light on how the 'SoA vs AoS' performance ratio changed over time. We compare the SoA performance to the maximal AoS performance (using the smallest possible particle size together with a stl vector). In Fig. 3 we compare Xeon CPUs from 2010 and 2014. For the two CPU generations SoA clearly wins over AoS. But the modern chip has a higher performance gain. Over only four years the gain has nearly doubled. The CPU architecture is more and more constructed in a way that favors the SoA layout.

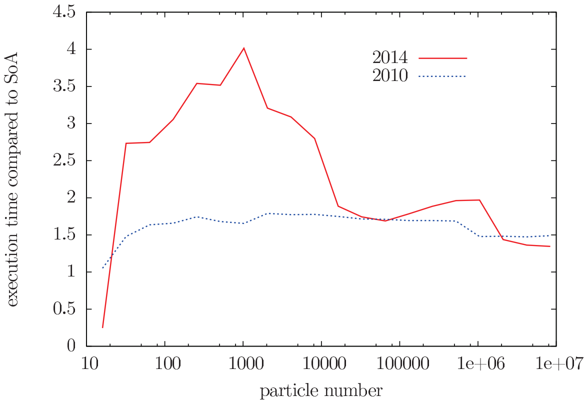


Figure 3: Benchmark comparing SoA and AoS for different CPU generations distinguished by the date of their commercial launch. 2014: Intel Xeon E5-2680 v3 (Haswell EP); 2010: Intel Xeon X5650 (Westmere EP)

One architectural component that has changed over the years is the performance of the vector unit. All today's CPUs possess so-called single instruction multiple data (SIMD) register and associated instruction sets. These allow to perform the same instruction (such as an addition) to many floating-point number at a time (in one cycle) that can significantly speed up code. In Fig. 4 we compare the performance of SoA and AoS with and without the use of the vector unit. The vectorization gain of a SoA reaches four to five for small particle numbers of the order of 100-1000 particles. The theoretical gain is eight as the used CPU has a 256 bit vector register containing eight single precision floating point values. At intermediate particle numbers (10^3-10^6) the gain is around two and vanishes for higher particle numbers. The origin of these regimes can be found in the three cache levels: The gain is maximal if all data fits into the L2 cache. The second regime corresponds to data fitting into the L3 cache. However, when the data size exceeds the latter the vectorization gain vanishes because the data has to be loaded from the main memory which is too slow to efficiently fill the vector registers.

Vectorization does not speed up AoS computations. Apparently, the auto-vectorizer of the compiler does not manage to create a substantial gain if a AoS is used. This means that a part of the SoA superiority can be explained by the fact that SoA effectively use the CPU vector units.

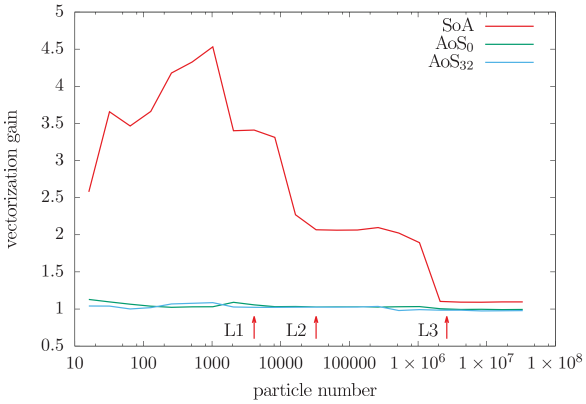


Figure 4: Performance gain due to vectorization for a SoA and AoSs.

This also explains the observed differences between the two CPU architectures. From one CPU generation to the other, the register width and the set of instruction has been augmented. The old CPU from 2010 has 128 bit vector register with a SSE4.2 instruction set and the most recent CPU from 2014 has a 256 bit vector register with an AVX2 instruction set. The factor of two between the 128 bit and 256 bit register explains the differences in Fig.3 for intermediate particle numbers. Of course other features than the vector unit changed among CPU architectures but it seems that most of the changes in the 'SoA vs AoS' performance ratio over the years are due to optimizations of the vector units.

3. Benchmarks on MICs and GPUs

Today's supercomputer often use accelerators to speed up computationally intensive parts of numerical codes. Mainly two different accelerator types exist:

Intel recently introduced the 'many integrated core' (MIC) concept with the Xeon Phi co-processor that assembles many computing cores (around 60) on one chip. The used computing cores are simplified versions of commonly used CPUs so that numerical code compile without changes on a Xeon Phi.

Nvidia and AMD/ATI developed graphics processing units (GPU) that are now often used in high performance computing. This architecture uses hundreds to thousands of very simple computing cores to speed up high parallel algorithms. For these GPUs the numerical code has to be especially designed.

The importance of these accelerators for HPC is underlined by the fact that they are massively employed by the two fastest supercomputers in the world (according to the TOP 500 list, www.top500.org). In fact, Tianhe-2 uses Xeon Phi co-processors and Titan Nvidia GPUs.

3.1. MIC

During the last decade, the performance of supercomputers grew essentially by increasing the number of (standard) computing cores so that high performance computing demanded more and more for parallel numerical algorithms and codes. Intel pushes now further in the direction of massive parallel programming by introducing co-processors, called Xeon Phi, with around 60 integrated cores each. A single core is in general compatible to standard CPUs but exhibits some architectural differences that are important for the performance of SoAs and AoSs: A Xeon Phi has no L3 cache but only a 32 kByte L1 and a 512 kByte L2 cache per core. Another aspect is that the vectorization capacities have been improved by extending the SIMD registers to 512 bits which means that either 16 single precision floating point number or 8 double precision number can be processed in one cycle.

These design differences show up in the relative performance of AoS compared to SoA (as before, we will only study the single-core performance). Our benchmark shows the the MIC cores favor SoAs over AoS and that even more than standard CPUs. For small size objects and intermediate particle numbers the tested SoA is roughly ten times faster than the AoS (see Fig. 5). If the stored particle has a considerable size, this difference even varies between twenty and forty.

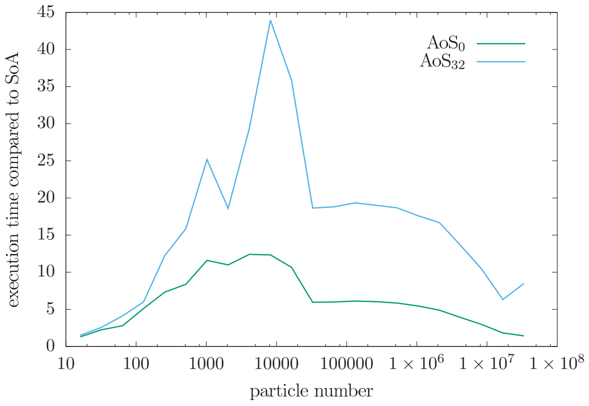


Figure 5: Benchmark comparing the performance of AoS to SoA on a Xeon Phi.

The reason is the extended vector performance of the MIC cores. Up to the point when the L2 cache is filled, vectorization speeds up the computation by a factor of roughly ten (see Fig.6) which is below the optimal value of sixteen but twice the speed-up measured for a standard CPU. Again, the cache size limits the particle number range for this speed-up.

3.2. GPU

The architecture behind the General Purpose Graphical Processing Units (GPU) uses a divide and conquer philosophy, by providing a many-core device, separated from the CPU, and typically connected to this one via a PCIe band. Graphic cards are widely used as accelerators in computer clusters, and power many of the TOP500 fastest supercomputer.

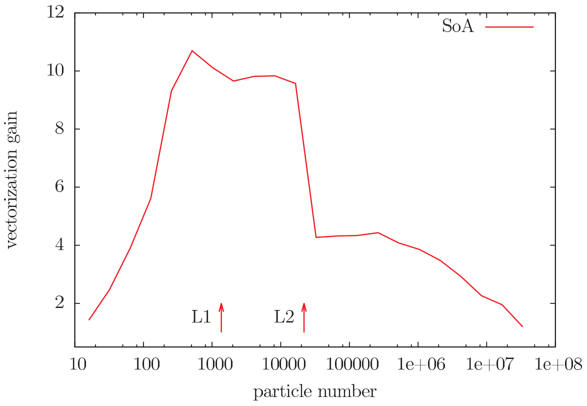


Figure 6: Speed-up by vectorization on a Xeon Phi

A few thousand of threads can run concurrently on the graphic card, thus providing the possibility to process many elements at a time. Furthermore, the architecture, labeled SIMT (for Single Instruction, Multiple Thread) is somewhat different from the SIMD in that every single thread has its own register state and can have independent behaviors from the others, a feature allowing a thread-based as well as coordinated threads development.

Another important difference from the CPU is the role of the L1 cache. Different caches co-exist, each one belonging to a given *streaming multiprocessor*, a structure responsible to dispatch the work among the threads. This cache is mainly used for register spilling and some stack variables. It does not promote temporal locality so that repeated operations on the same memory locations will not necessarily benefit from this cache. The L2 cache, shared among all streaming multiprocessors, will be used instead. We thus expect the SoA pattern not to benefit from the L1 cache, but the AoS will in fact benefit from it : indeed, loading a large structure into memory allows threads to reuse close memory.

A benchmark similar to those listed above is performed. The graphic card used is a Nvidia Tesla M2050, a middle-range, widespread computing device. The card has 448 cores, spread among 14 multiprocessors and the L2 cache size is $\sim$ 786 kBytes. In the SOA algorithm, three functions are launched, one per position and velocity component, with a number of threads such that each thread has a single element to process. The program is compiled with optimization. Timing is measured by the Nvidia profiling tool, allowing to isolate the kernel execution time from the overhead of the function calls. Execution times normalized by the number of particles are shown in Fig. 7. For large particle numbers, SoA outperforms AoS solution, by a factor ~ 2 for $SIZE = 0$ and ~ 20 for $SIZE = 32$. As the particle number decreases however, AoS performs better, with higher crossover for lower $SIZE$. The reason for this lies in the GPU architecture, as we will now explain.

Figure 9 displays two relevant measured metrics for the function used. The major drawback of the AoS approach is the well know effect of uncoalesced memory access, hence threads fetch unneeded data in the cache lines. This is particularly damageable in the case of GPU computing because the major weak point is the latency of memory access. Accessing data is done by a single, indivisible group of 32 threads, called a *warp*. Loading a large structure in a thread memory, only to read a small part of it, degrades badly the memory access performance up to a factor of 32. The case AoS with $SIZE = 0$ packs 6 values and will then have a memory performance of $1/6 \approx 16\%$ compared to SoA, and the highest values of $SIZE$ will display a performance down to $1/32 \approx 3\%$. This is shown in Fig. 8. As a result, one can clearly see that the performance per particle saturates for a sufficiently large number of particles, with SoA pattern outperforming the AoS with $SIZE = 32$ by a factor of 20 and the AoS with $SIZE = 0$ by a factor of 2. For small particle numbers, performance is hindered by a less effective usage of memory, additional to the uncoalesced access pattern, as can be seen in Figure 8.

It is also noticeable that the performance of SoA is slightly worse than AoS for small particle numbers (up to 1000). This can be attributed to the fact that when the number of particles is small enough, the L1 cache and the threads registers are large enough to keep the whole particles close in memory, hence allowing faster access to other position and velocity components for successive operations, while the SoA pattern has to make a request to global memory for every needed data. Nevertheless, this effect only brings advantage when the particle number is small. When this number increases, the cache cannot hold the data anymore and so that the global memory is used and another long latency fetch has to be performed. The caching advantage is thus eventually taken over by the poor memory access performance, and the crossing between SoA and AoS (with $SIZE = 0$) occurs around 2000 particles. This corresponds to a full utilisation of the L1 cache which is 48 kB, the size of one SoA ($SIZE = 0$) particle being $6 * 4 = 24$ bytes.

We also performed this simple benchmark on another multipurpose graphic card boarded on a desktop computer. For this example, we used the Nvidia Geforce GT755M, composed of 384 cores on 2 multiprocessors, with ~ 262 kbytes. Timings were $2 - 3$ times slower, irrespective of the number of particles and both for AoS and SoA (not shown here), illustrating the benefit of using a graphic card specifically dedicated to high performance computing exhibiting more parallelism.

4. Generic C++ implementation of a structure of arrays (SoAx)

In the introduction we have seen that implementing, maintaining and using a structure of array can be annoying. We present now an implementation of a structure of array using modern C++ (in fact C++11), called SoAx, that provides a handy interface, high flexibility and optimal performance. We use C++ because is enables powerful mechanisms to build abstractions without loss of performance. We discussed in the introduction that adding a property (such as a charge) to a particle requires the modification of all member functions (such as PartArr::allocate) that handle the different arrays. C++ allows to pass this task to the compiler. Using *template meta programming* [6], the needed code can be automatically generated during the compilation. The result is a class that contains an array for each particle property, the associated access functions and member functions that allow efficient handling of all arrays.

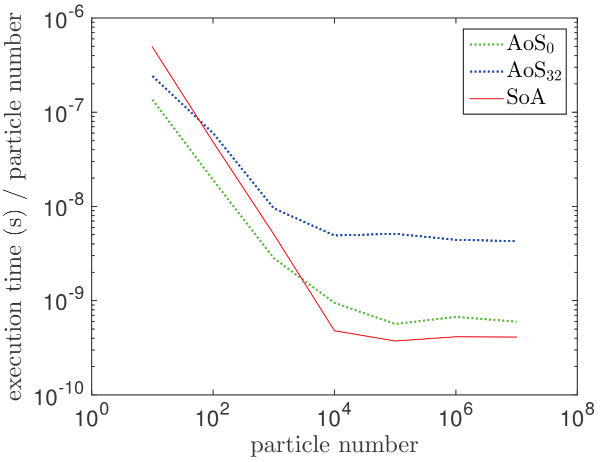


Figure 7: Benchmark comparing the execution time of AoS vs SoA implementation of the Eulerian update step in single precision.

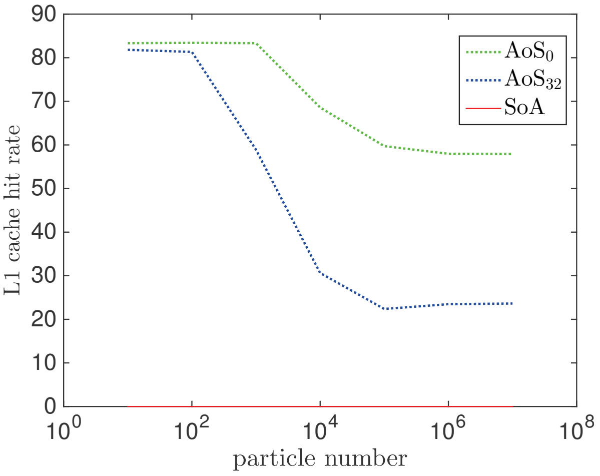


Figure 8: L1 cache hit rate for global memory load requests, in percents.

4.1. Using SoAx

Before discussing details of the implementation let us first show a short listing presenting some functionality of SoAx. Let us assume that we want our particles to have an identity, a position, a velocity, and a mass of types `int`, `double`, `double`, and

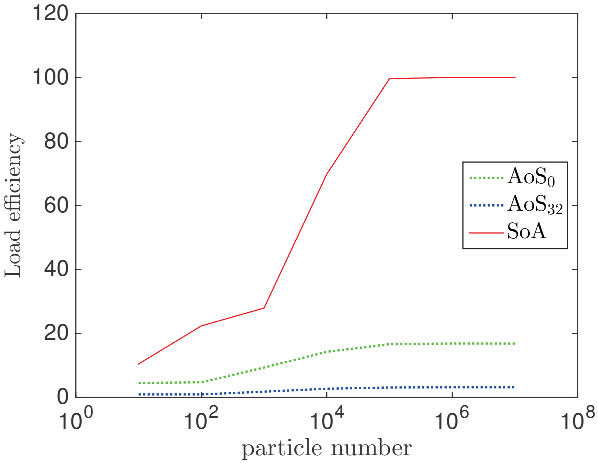


Figure 9: Loading efficiency from the main GPU memory. This is the ratio between requested memory and effectively used memory.

`float`, respectively. Let us further assume that we need three-dimensional coordinates for the position and velocity. Here is what one could write using SoAx:

Listing 5: Example code showing typical usage of SoAx

```
// Define particle properties through macro
SOAX_ATTRIBUTE(id, 'N');        // identity
SOAX_ATTRIBUTE(pos, 'P');       // position
SOAX_ATTRIBUTE(vel, 'V');       // velocity
SOAX_ATTRIBUTE(mass, 'M');      // velocity

// Specify types and dimension and
// concatenate attributes using std::tuple
typedef std::tuple<id<int,1>,
                   pos<double,3>,
                   vel<double,3>,
                   mass<float,1>> ArrayTypes;

// create SoA for 42 particles
Soax<ArrayTypes> soax(42);

// access properties of particle 23
soax.id(23) = 0;          // set identity
soax.pos(23,0) = 100;     // set x-coordinate

// operations on all particles (x=v_y-v_z)
soax.posArr(0) =soax.velArr(1)-soax.velArr(2);

// allocate memory of 100 particles
soax.resize(100);
```

We have payed attention to the fact, that user might want to extract and treat particles as objects (in the spirit of `struct Particle`). With SoAx one can write

Listing 6: Example of using SoAx elements

```
  auto particle = soax.getElement(7);
  particle.id() = 42;
  particle.pos(0) = 3.14;
  soax.push_back(particle);
```

The necessary class from which the particle objects are created is also automatically created by the compiler by means of template meta programming. This technique will be discussed in the next section.

4.2. Implementation of SoAx

SoAx uses *inheritance* in combination with *template meta-programming*. The basic idea is to inherit all arrays (particle properties) into one single structure. The different property types of the particle are passed to the SoAx class using std::tuple. This is a component of C++11 storing heterogeneous data types.

A SoAx attribute consists of an array for storing and member-functions for accessing data. We have chosen to generate these attribute classes by macros to avoid repetitive implementations as they have all the same structure. Macros permit to give custom names to the attributes: From `SOAX_ATTRIBUTE(pos, 'P');` the compiler creates a class with a member-function `pos` to access individual particles and `posArr` to access directly the complete array. The character `P` is only a descriptive string that can be used by the user for other purposes. `pos<double,N>` is an instantiation of the class template holding a N-dimensional array of type `double`.

Let us here mention that advanced programming techniques can be used to provide usage safety. The dimensionality is for example automatically taken into account for the member function `pos`. In the case of `pos<double,3>`, `pos(42,0)` gives the expected access to the first coordinate of particle 42 while `pos(42)` yields a compile-time assertion (through 'substitution failure is not an error' (SFINAE, [7])). The behavior is the opposite in the case of `id<int,1>`, where `id(42)` is the identity of particle 42 and `id(42,0)` results in a compile-time assertion.

Advanced programming techniques also allow to enable the library user to write automatically optimized code. The line `soax.posArr(0) = soax.velArr(1)-soax.velArr(2);` in List. 5 performs an operation on all particles. The library user does not need to write a custom for-loop for CPUs or a CUDA kernel for GPUs. For this, SoAx uses a technique called *expression templates* [8, 7] where a computation such as a sum is encoded in a template. Chained arithmetic operations are analysed at compile time and an optimized code without unnecessary copies is generated by the compiler. This technique is nowadays used in linear algebra software [9].

4.2.1. Adding functions

The user can easily add custom functions to SoAx that he wants to be applied to all arrays. For this, it is not necessary to touch the code of the library. The user only has to define a structure containing a `doIt` member-function (see List. 7 for an example). The first template parameter of this member `doIt` is a reference to one of the SoA arrays. Other parameters can be freely chosen (internally SoAx uses variadic templates). Here is an example of a function that sets the values of all arrays to a certain value:

Listing 7: Example of a function to be applied to all SoAx arrays

```
struct SetToValue
{
  template<class T, class Type>
  static void doIt(T& t, Type value) {
    for(int i=0;i<t->size();i++)
    t->operator[](i) = value;
  }
};
```

Passing this function to a SoAx object `soax` as a template argument, `soax.apply<SOAX::SetToValue>(42);` applies `SetToValue::doIt` to all arrays in `soax`.

This is achieved via recursive templates. We discuss this programming technique here as a showcase for the `doIt` function as it explains how templates can be used to make the compiler generate code without loss of performance (see List. 8). In fact, the SoAx member-function `apply` calls the member-function `doIt` of the class template `TupleDo` with the particle attribute tuple (`Tuple`), its size (`N`) and the user defined template (`DoItClass` = e.g. `SetToValue`) as template arguments. The member-function `doIt` calls recursively `TupleDo::doIt` for the attribute tuple but passing a decremented size. This recursion continues until the passed size is one so that the compiler chooses the partially specialized case below. Its `doIt` member-function calls the `doIt` function of the user provided `DoItClass` that terminates the treatment of the first entry of the attribute tuple `Tuple`. After that the `DoItClass::doIt` is called for the second entry. This process continues for all attributes. As the code for all calls is generated at compile time, there is no performance overhead compared to a hand-written code.

Listing 8: Example explaining compile time code creation by recursive templates

```
template<class Tuple, std::size_t N, class
    DoItClass>
struct TupleDo {

  template<class... Args>
  static void doIt(Tuple& t, Args... args)
  {
    TupleDo<Tuple, N-1, DoItClass>::doIt(t,
        args...);
    DoItClass::doIt(std::get<N-1>(t),args...);
  }
};

template<class Tuple, class DoItClass>
struct TupleDo<Tuple, 1, DoItClass> {

  template<class... Args>
  static void doIt(Tuple& t, Args... args)
  {
     DoItClass::doIt(std::get<0>(t),args...);
  }
};
```

4.2.2. GPU implementation

Several restrictions apply when working with GPU processors. A first one is the costly data transfer between CPU and GPU: one has to design a solution in which those transfers are minimized. Data should reside mainly on the GPU and be transferred to the main memory only when needed by the CPU, for example for output to a hard drive. One thus cannot make use of solutions that would results in dereferenciation by the CPU of each elements one at a time, but must rely on device functions that process all data at once on the device. In addition, when processing multiple vectors with several operations, processing them all together is faster than successively, an optimisation sometimes referred to as *loop fusion*. These constraits lead us to make again use of expression templates for device data.

Another constrain comes from the fact that C++-stl vectors are not designed to work on GPU processors within the CUDA framework, as far as the version 7.0, and another type of data storage is then needed. To allow expression templates to work with GPUs, we build a custom class, called deviceWrapper, encompassing a pointer to data living on the device. In addition, as the THRUST library provides the best mimic of stl vectors structure and algorithms to our knowledge, we also keep trace of the associated device vector to allow efficient operations to be performed on the data.

When an assignment (of the form `soax.posArr(0) = soax.velArr(1)-soax.velArr(2);`) is performed, a kernel is called and passed a copy of the underlying deviceWrapper object, accessing the data with the expression template objects. The copy constructor of the deviceWrapper class then needs to be overloaded in order to copy *only* the raw device pointer and not all the data at each call.

Fig. 10 shows a benchmark evaluating the performance of this implementation for the operation (1) as a function of the particle number, along with the SOA and AOS (with $SIZE = 16$) implementations as references. The time is measured this time with a std::chrono rather then with the kernel profiler, allowing to assess the possible overhead of the SoAx solution. With this benchmark, we confirm that the performance of SoAx is the same as the SOA also on GPUs. Indeed, the SoAx GPU implementation comes down *in fine* to call a kernel on the stored data adressed through expression templates.

5. Conclusions

The goal of the work is two-fold. First, it shows that heterogeneous data (such as particles) should be implemented in an *array of structure* (AoS) fashion rather than in a *structure of array* (SoA) one if performance is crucial. AoS are generally much faster on modern CPUs as well as on GPUs. The reason is that AoS better uses cache and vectorization resources that can speed up typical number crunching algorithms on particles by more than one order of magnitude. However, implementing and maintaining AoS can be cumbersome especially if the the number of numerical types representing a particle change from one application to another. SoA are in general more handy and

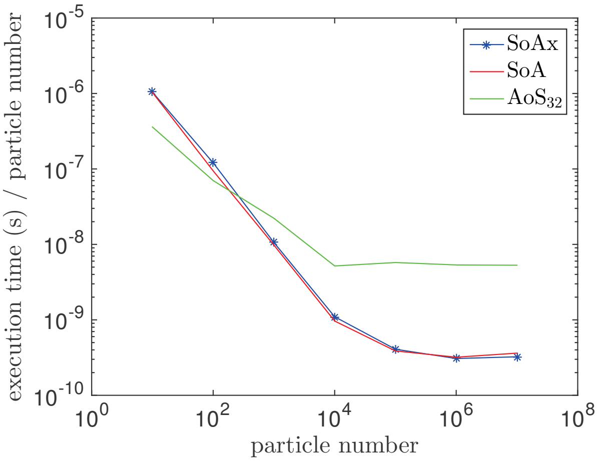


Figure 10: Benchmark of SoAx library GPU implementation.

flexible. This consideration leads to the second contribution of this work showing that modern C++ programming techniques permits to combine the advantages of both concepts (SoA and AoS) to build a generic library that has the performance of SoAs and the flexibility and handiness of AoS. We demonstrate the benefit of template meta programming for scientific codes. This technique delegates code generation to the compiler and allows for highly readable, maintainable and fast application code. The presented library SoAx runs on CPUs as well as on GPUs.

Acknowledgment

We thank K. Thust for useful discussion and his help using the MIC co-processor. We also thank A. Miniussi for fruitful advices concerning Template Meta Programming. Access to supercomputer Jureca and Juropa3 at the FZ Jülich was made available through project HBO22. Part of the computations were performed on the 'mesocentre de calcul SIGAMM' in Nice.

References

[1] R. Teyssier, Cosmological hydrodynamics with adaptive mesh refinement: a new high-resolution code called ramses, A & A 385 (2002) 337.

[2] K. Germaschewski, W. Fox, S. Abbott, N. Ahmadi, K. Maynard, L. Wang, H. Ruhl, A. Bhattacharjee, The plasma simulation code: A modern particle-in-cell code with load-balancing and gpu support.

[3] L. Biferale, G. Boffetta, A. Celani, B. J. Devenish, A. Lanotte, F. Toschi, Multifractal statistics of lagrangian velocity and acceleration in turbulence, Phys. Rev. Lett. 93 (6) (2004) 4502.

[4] C. Huang, B. Shi, N. He, Z. Chai, Implementation of multi-gpu based lattice boltzmann method for flow through porous media, Advances in Applied Mathematics and Mechanics 7 (01) (2015) 1–12.

[5] W. Xue, C. Yang, H. Fu, X. Wang, Y. Xu, J. Liao, L. Gan, Y. Lu, R. Ranjan, L. Wang, Ultra-scalable cpu-mic acceleration of mesoscale atmospheric modeling on tianhe-2, IEEE Transactions on Computers 64 (8) (2015) 2382–2393.

[6] D. Abrahams, A. Gurtovoy, C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond, Addison-Wesley Professional, 2005.

[7] D. Vandevoorde, N. Josuttis, C++ Templates: The Complete Guide, Addison-Wesley Professional, 2002.

[8] T. Veldhuizen, Expression templates, C++ Report 7 (1995) 26–32.

[9] A. M. Aragón, A c++11 implementation of arbitrary-rank tensors for high-performance computing, Computer Physics Communications 185 (2014) 1681 – 1696.

Bibliography

ABRAHAMSON, J 1975 Collision rates of small particles in a vigorously turbulent fluid. *Chem. Eng. Sci.* **30** (11), 1371–1379.

AGUINAGA, SYLVAIN, SIMONIN, OLIVIER, BORÉE, JACQUES & HERBERT, VINCENT 2009 A simplified particle-turbulence interaction pdf model: application to deposition modelling in turbulent boundary layer. In *ASME 2009 Fluids Engineering Division Summer Meeting*, pp. 1899–1910. American Society of Mechanical Engineers.

ARMITAGE, PHILIP J 2015 Physical processes in protoplanetary disks. *arXiv preprint arXiv:1509.06382* .

AYALA, ORLANDO, GRABOWSKI, WOJCIECH W & WANG, LIAN-PING 2007 A hybrid approach for simulating turbulent collisions of hydrodynamically-interacting particles. *Journal of Computational Physics* **225** (1), 51–73.

BALACHANDAR, S & EATON, JOHN K 2010 Turbulent dispersed multiphase flow. *Annual Review of Fluid Mechanics* **42**, 111–133.

BALKOVSKY, E, FALKOVICH, G & FOUXON, A 2001 Intermittent distribution of inertial particles in turbulent flows. *Physical Review Letters* **86** (13), 2790.

BALKOVSKY, E & LEBEDEV, V 1998 Instanton for the kraichnan passive scalar problem. *Physical Review E* **58** (5), 5776.

BARDOS, CLAUDE & TADMOR, EITAN 2015 Stability and spectral convergence of fourier method for nonlinear problems: on the shortcomings of the 2/3 de-aliasing method. *Numerische Mathematik* **129** (4), 749–782.

Barranco, Joseph A & Marcus, Philip S 2005 Three-dimensional vortices in stratified protoplanetary disks. *The Astrophysical Journal* **623** (2), 1157.

Batchelor, GK 1950 The application of the similarity theory of turbulence to atmospheric diffusion. *Quarterly Journal of the Royal Meteorological Society* **76** (328), 133–146.

Bec, Jeremie 2003 Fractal clustering of inertial particles in random flows. *Physics of Fluids (1994-present)* **15** (11), L81–L84.

Bec, Jérémie 2005 Multifractal concentrations of inertial particles in smooth random flows. *Journal of Fluid Mechanics* **528**, 255–277.

Bec, J, Celani, Antonio, Cencini, M & Musacchio, S 2005 Clustering and collisions of heavy particles in random smooth flows. *Physics of Fluids (1994-present)* **17** (7), 073301.

Beigie, Darin, Leonard, Anthony & Wiggins, Stephen 1994 Invariant manifold templates for chaotic advection. *Chaos, Solitons & Fractals* **4** (6), 749–868.

Bell, Nathan & Hoberock, Jared 2011 Thrust: A productivity-oriented library for cuda. *GPU computing gems Jade edition* **2**, 359–371.

Bergougnoux, Laurence, Bouchet, Gilles, Lopez, Diego & Guazzelli, Elisabeth 2014 The motion of solid spherical particles falling in a cellular flow field at low stokes number. *Phys. Fluids* **26** (9), 093302.

Bernard, Denis 2000 Influence of friction on the direct cascade of the 2d forced turbulence. *EPL (Europhysics Letters)* **50** (3), 333.

Bhatnagar, Akshay, Gupta, Anupam, Mitra, Dhrubaditya, Pandit, Rahul & Perlekar, Prasad 2016 How long do particles spend in vortical regions in turbulent flows? *Physical Review E* **94** (5), 053119.

Bhatnagar, Prabhu Lal, Gross, Eugene P & Krook, Max 1954 A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical review* **94** (3), 511.

Bianchi, Stefano, Biferale, Luca, Celani, Antonio & Cencini, Massimo 2016 On the evolution of particle-puffs in turbulence. *European Journal of Mechanics-B/Fluids* **55**, 324–329.

Bodenschatz, Eberhard, Malinowski, Szymon P, Shaw, Raymond A & Stratmann, Frank 2010 Can we understand clouds without turbulence? *Science* **327** (5968), 970–971.

Boer, GJ, McFarlane, NA, Laprise, R, Henderson, JD & Blanchet, J-P 1984 The canadian climate centre spectral atmospheric general circulation model. *Atmosphere-Ocean* **22** (4), 397–429.

Boffetta, G & Celani, A 2000 Pair dispersion in turbulence. *Physica A: Statistical Mechanics and its Applications* **280** (1), 1–9.

Boffetta, G, Celani, A, De Lillo, F & Musacchio, S 2007 The eulerian description of dilute collisionless suspension. *EPL (Europhysics Letters)* **78** (1), 14001.

Boffetta, Guido, Celani, Antonio, Musacchio, S & Vergassola, Massimo 2002 Intermittency in two-dimensional ekman-navier-stokes turbulence. *Physical Review E* **66** (2), 026304.

Boffetta, G, Celani, A & Vergassola, M 2000 Inverse energy cascade in two-dimensional turbulence: Deviations from gaussian behavior. *Physical Review E* **61** (1), R29.

Boffetta, G, Cenedese, A, Espa, Stefania & Musacchio, S 2005 Effects of friction on 2d turbulence: An experimental study of the direct cascade. *EPL (Europhysics Letters)* **71** (4), 590.

Boffetta, G, De Lillo, F & Gamba, A 2004 Large scale inhomogeneity of inertial particles in turbulent flows. *Physics of Fluids (1994-present)* **16** (4), L20–L23.

Boffetta, Guido & Ecke, Robert E 2012 Two-dimensional turbulence. *Annual Review of Fluid Mechanics* **44**, 427–451.

Boffetta, G & Musacchio, S 2010 Evidence for the double cascade scenario in two-dimensional turbulence. *Physical Review E* **82** (1), 016307.

Borue, Vadim 1994 Inverse energy cascade in stationary two-dimensional homogeneous turbulence. *Physical review letters* **72** (10), 1475.

Bourgoin, Mickaël 2015 Turbulent pair dispersion as a ballistic cascade phenomenology. *Journal of Fluid Mechanics* **772**, 678–704.

Brigham, E Oran & Brigham, Elbert Oran 1974 *The fast Fourier transform*, , vol. 7. Prentice-Hall Englewood Cliffs, NJ.

Bryan, Greg L, Norman, Michael L, O'Shea, Brian W, Abel, Tom, Wise, John H, Turk, Matthew J, Reynolds, Daniel R, Collins, David C, Wang, Peng, Skillman, Samuel W *et al.* 2014 Enzo: An adaptive mesh refinement code for astrophysics. *The Astrophysical Journal Supplement Series* **211** (2), 19.

Cantwell, Brian J 1993 On the behavior of velocity gradient tensor invariants in direct numerical simulations of turbulence. *Physics of Fluids A: Fluid Dynamics (1989-1993)* **5** (8), 2008–2013.

Canuto, Claudio, Hussaini, M Yousuff, Quarteroni, Alfio Maria, Thomas Jr, A *et al.* 2012 *Spectral methods in fluid dynamics*. Springer Science & Business Media.

Celani, Antonio, Afonso, Marco Martins & Mazzino, Andrea 2007 Point-source scalar turbulence. *Journal of Fluid Mechanics* **583**, 189–198.

Celani, A, Lanotte, A, Mazzino, A & Vergassola, M 2000 Universality and saturation of intermittency in passive scalar turbulence. *Physical review letters* **84** (11), 2385.

Celani, A, Lanotte, A, Mazzino, A & Vergassola, M 2001 Fronts in passive scalar turbulence. *Physics of Fluids (1994-present)* **13** (6), 1768–1783.

Celani, A & Vergassola, M 2001 Statistical geometry in scalar turbulence. *Physical review letters* **86** (3), 424.

Celani, Antonio, Villermaux, Emmanuel & Vergassola, Massimo 2014 Odor landscapes in turbulent environments. *Physical Review X* **4** (4), 041015.

Cercignani, Carlo 1988 The boltzmann equation. In *The Boltzmann Equation and Its Applications*, pp. 40–103. Springer.

Chen, L, Goto, S & Vassilicos, JC 2006*a* Turbulent clustering of stagnation points and inertial particles. *Journal of Fluid Mechanics* **553**, 143–154.

Chen, Shiyi, Ecke, Robert E, Eyink, Gregory L, Rivera, Michael, Wan, Minping & Xiao, Zuoli 2006*b* Physical mechanism of the two-dimensional inverse energy cascade. *Physical review letters* **96** (8), 084502.

Cheng, John, Grossman, Max & McKercher, Ty 2014 *Professional Cuda C Programming*. John Wiley & Sons.

Chibbaro, Sergio & Minier, Jean-Pierre 2008 Langevin pdf simulation of particle deposition in a turbulent pipe flow. *Journal of aerosol science* **39** (7), 555–571.

Chibbaro, Sergio & Minier, Jean-Pierre 2011 A note on the consistency of hybrid eulerian/lagrangian approach to multiphase flows. *International Journal of Multiphase Flow* **37** (3), 293–297.

Corrsin, Stanley 1975 Limitations of gradient transport models in random walks and in turbulence. *Advances in geophysics* **18**, 25–60.

Cushman-Roisin, Benoit & Beckers, Jean-Marie 2011 *Introduction to geophysical fluid dynamics: physical and numerical aspects*, , vol. 101. Academic Press.

Daniel, W Brent & Rutgers, Maarten A 2002 Topology of two-dimensional turbulence. *Physical review letters* **89** (13), 134502.

Derevyanko, SA, Falkovich, G, Turitsyn, K & Turitsyn, S 2007 Lagrangian and eulerian descriptions of inertial particles in random flows. *Journal of Turbulence* (8), N16.

Deusebio, Enrico, Boffetta, Guido, Lindborg, Erik & Musacchio, Stefano 2014 Dimensional transition in rotating turbulence. *Physical Review E* **90** (2), 023005.

Drew, Donald A & Passman, Stephen L 1999 *Theory of multicomponent fluids*, , vol. 135. Springer Science & Business Media.

Dubos, Thomas & Babiano, Armando 2003 Comparing the two-dimensional cascades of vorticity and a passive scalar. *Journal of Fluid Mechanics* **492**, 131–145.

Eckmann, J-P & Ruelle, David 1985 Ergodic theory of chaos and strange attractors. *Reviews of modern physics* **57** (3), 617.

Eswaran, V & Pope, SB 1988 An examination of forcing in direct numerical simulations of turbulence. *Computers & Fluids* **16** (3), 257–278.

Fede, Pascal, Sofonea, Victor, Fournier, Richard, Blanco, Stéphane, Simonin, Olivier, Lepoutère, Guillaume & Ambruş, Victor 2015 Lattice boltzmann model for predicting the deposition of inertial particles transported by a turbulent flow. *International Journal of Multiphase Flow* **76**, 187–197.

Fjørtoft, Ragnar 1953 On the changes in the spectral distribution of kinetic energy for twodimensional, nondivergent flow. *Tellus* **5** (3), 225–230.

Fox, Rodney O 2012 Large-eddy-simulation tools for multiphase flows. *Annual Review of Fluid Mechanics* **44**, 47–76.

Fox, Rodney O, Laurent, Frédérique & Massot, Marc 2008 Numerical simulation of spray coalescence in an eulerian framework: direct quadrature method of moments and multi-fluid method. *Journal of Computational Physics* **227** (6), 3058–3088.

Frisch, Uriel 1995 *Turbulence: the legacy of AN Kolmogorov*. Cambridge university press.

Ge, Wei, Chen, Feiguo, Gao, Jian, Gao, Shiqiu, Huang, Jin, Liu, Xiaoxing, Ren, Ying, Sun, Qicheng, Wang, Limin, Wang, Wei *et al.* 2007 Analytical multi-scale method for multi-phase complex systems in process engineering—bridging reductionism and holism. *Chemical Engineering Science* **62** (13), 3346–3377.

Gillespie, Daniel T 1996 Exact numerical simulation of the ornstein-uhlenbeck process and its integral. *Physical review E* **54** (2), 2084.

Girimaji, Sharath S & Zhou, Ye 1996 Analysis and modeling of subgrid scalar mixing using numerical data. *Physics of Fluids (1994-present)* **8** (5), 1224–1236.

Godoy-Diana, Ramiro, Chomaz, Jean-Marc & Billant, Paul 2004 Vertical length scale selection for pancake vortices in strongly stratified viscous fluids. *Journal of Fluid Mechanics* **504**, 229–238.

Goto, Susumu & Vassilicos, JC 2006 Self-similar clustering of inertial particles and zero-acceleration points in fully developed two-dimensional turbulence. *Physics of Fluids (1994-present)* **18** (11), 115103.

Grabowski, Wojciech W & Wang, Lian-Ping 2013 Growth of cloud droplets in a turbulent environment. *Annual review of fluid mechanics* **45**, 293–324.

Grassberger, Peter & Procaccia, Itamar 2004 Measuring the strangeness of strange attractors. In *The Theory of Chaotic Attractors*, pp. 170–189. Springer.

Gualtieri, Paolo, Picano, F, Sardina, Gaetano & Casciola, Carlo Massimo 2015 Exact regularized point particle method for multiphase flows in the two-way coupling regime. *Journal of Fluid Mechanics* **773**, 520–561.

Habich, Johannes, Zeiser, Thomas, Hager, Georg & Wellein, Gerhard 2011 Performance analysis and optimization strategies for a d3q19 lattice boltzmann kernel on nvidia gpus using cuda. *Advances in Engineering Software* **42** (5), 266–272.

Han, Xianglu & Naeher, Luke P 2006 A review of traffic-related air pollution exposure assessment studies in the developing world. *Environment international* **32** (1), 106–120.

Harlow, Francis H 2004 Fluid dynamics in group t-3 los alamos national laboratory:(la-ur-03-3852). *Journal of Computational Physics* **195** (2), 414–433.

Hentschel, HGE & Procaccia, Itamar 1983 The infinite number of generalized dimensions of fractals and strange attractors. *Physica D: Nonlinear Phenomena* **8** (3), 435–444.

Van der Hoef, MA, van Sint Annaland, M, Deen, NG & Kuipers, JAM 2008 Numerical simulation of dense gas-solid fluidized beds: a multiscale modeling strategy. *Annu. Rev. Fluid Mech.* **40**, 47–70.

Honeycutt, Rebecca L 1992*a* Stochastic runge-kutta algorithms. i. white noise. *Physical Review A* **45** (2), 600.

Honeycutt, Rebecca L 1992*b* Stochastic runge-kutta algorithms. ii. colored noise. *Physical Review A* **45** (2), 604.

Hundsdorfer, Willem, Koren, Barry, Verwer, JG *et al.* 1995 A positive finite-difference advection scheme. *Journal of computational physics* **117** (1), 35–46.

Hunt, Julian CR 1998 Lewis fry richardson and his contributions to mathematics, meteorology, and models of conflict. *Annual Review of Fluid Mechanics* **30** (1), xiii–xxxvi.

Hwang, Wontae & Eaton, John K 2006 Homogeneous and isotropic turbulence modulation by small heavy ($st \sim 50$) particles. *Journal of Fluid Mechanics* **564**, 361–393.

Iglberger, Klaus, Hager, Georg, Treibig, Jan & Rüde, Ulrich 2012 Expression templates revisited: a performance analysis of current methodologies. *SIAM Journal on Scientific Computing* **34** (2), C42–C69.

Jeffreys, Harold & Jeffreys, Bertha 1999 *Methods of mathematical physics*. Cambridge university press.

Johansen, Anders, Oishi, Jeffrey S, Mac Low, Mordecai-Mark, Klahr, Hubert, Henning, Thomas & Youdin, Andrew 2007 Rapid planetesimal formation in turbulent circumstellar disks. *Nature* **448** (7157), 1022–1025.

Kaplan, JL & Yorke, JA 1979 Functional differential equations and approximation of fixed points. *Lecture notes in mathematics* **730**, 228.

Kellay, H, Wu, XL & Goldburg, WI 1998 Vorticity measurements in turbulent soap films. *Physical review letters* **80** (2), 277.

Kolmogorov, Andrey Nikolaevich 1941 Dissipation of energy in locally isotropic turbulence. In *Dokl. Akad. Nauk SSSR*, , vol. 32, pp. 16–18. JSTOR.

Koren, Barry 1993 *A robust upwind discretization method for advection, diffusion and source terms*. Centrum voor Wiskunde en Informatica Amsterdam.

Kraichnan, R H 1967 Inertial ranges in two-dimensional turbulence. *Physics of Fluids* .

Kraichnan, Robert H 1971 Inertial-range transfer in two-and three-dimensional turbulence. *Journal of Fluid Mechanics* **47** (03), 525–535.

Kraichnan, Robert H 1994 Anomalous scaling of a randomly advected passive scalar. *Physical review letters* **72** (7), 1016.

Kraichnan, Robert H & Montgomery, David 1980 Two-dimensional turbulence. *Reports on Progress in Physics* **43** (5), 547.

KRSTULOVIC, GIORGIO, CENCINI, MASSIMO & BEC, JEREMIE 2013 Effective rates in dilute reaction-advection systems for the annihilation process a + a. *Journal of Statistical Physics* **153** (3), 530–550.

LABOURASSE, EMMANUEL, LACANETTE, DELPHINE, TOUTANT, A, LUBIN, P, VINCENT, S, LEBAIGUE, O, CALTAGIRONE, J-P & SAGAUT, P 2007 Towards large eddy simulation of isothermal two-phase flows: governing equations and a priori tests. *International journal of multiphase flow* **33** (1), 1–39.

LAENEN, FRANÇOIS, KRSTULOVIC, GIORGIO & BEC, JÉRÉMIE 2016 A lattice method for the eulerian simulation of heavy particle suspensions. *Comptes Rendus Mécanique* .

LANDAU, LD & LIFSHITZ, EM 1987 Fluid mechanics. 1987. *Course of Theoretical Physics* .

LAPEYRE, G & KLEIN, P 2006 Dynamics of the upper oceanic layers in terms of surface quasigeostrophy theory. *Journal of physical oceanography* **36** (2), 165–176.

LAURENT, FRÉDÉRIQUE & MASSOT, MARC 2001 Multi-fluid modelling of laminar polydisperse spray flames: origin, assumptions and comparison of sectional and sampling methods. *Combustion theory and modelling* **5** (4), 537–572.

LAURENT, FRÉDÉRIQUE, MASSOT, MARC & VILLEDIEU, PHILIPPE 2004 Eulerian multi-fluid modeling for the numerical simulation of coalescence in polydisperse dense liquid sprays. *Journal of Computational Physics* **194** (2), 505–543.

LEVEQUE, RANDALL J 2002 *Finite volume methods for hyperbolic problems*, , vol. 31. Cambridge university press.

LIN, JC, BRUNNER, D & GERBIG, C 2011 Studying atmospheric transport through lagrangian models. *Eos* **92** (21), 177–178.

LYAPUNOFF, A 1907 Probléme günüral de la stabilitü du mouvement. *Annales de la Facultü des Sciences de Toulouse* **9**, 209.

MAFRA-NETO, AGENOR, CARDÉ, RING T *et al.* 1994 Fine-scale structure of pheromone plumes modulates upwind orientation of flying moths. *Nature* **369** (6476), 142–144.

MAJDA, ANDREW J & KRAMER, PETER R 1999 Simplified models for turbulent diffusion: theory, numerical modelling, and physical phenomena. *Physics reports* **314** (4), 237–574.

MANN, KENNETH HENRY & LAZIER, JOHN RN 2013 *Dynamics of marine ecosystems: biological-physical interactions in the oceans*. John Wiley & Sons.

MANNEVILLE, PAUL 2010 *Instabilities, chaos and turbulence*, , vol. 1. World Scientific.

Maxey, MR 1987 The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields. *Journal of Fluid Mechanics* **174**, 441–465.

Maxey, Martin R & Riley, James J 1983 Equation of motion for a small rigid sphere in a nonuniform flow. *Physics of Fluids (1958-1988)* **26** (4), 883–889.

Meheut, H, Meliani, Z, Varniere, P & Benz, W 2012 Dust-trapping rossby vortices in protoplanetary disks. *Astronomy & Astrophysics* **545**, A134.

Mignone, A, Zanni, C, Tzeferacos, P, van Straalen, B, Colella, P & Bodo, G 2011 The pluto code for adaptive mesh computations in astrophysical fluid dynamics. *The Astrophysical Journal Supplement Series* **198** (1), 7.

Minier, Jean-Pierre & Peirano, Eric 2001 The pdf approach to turbulent polydispersed two-phase flows. *Physics Reports* **352** (1), 1–214.

Mininni, Pablo D, Rosenberg, Duane, Reddy, Raghu & Pouquet, Annick 2011 A hybrid mpi–openmp scheme for scalable parallel pseudospectral computations for fluid turbulence. *Parallel Computing* **37** (6), 316–326.

Monahan, Sarah M & Fox, Rodney O 2007 Effect of model formulation on flow-regime predictions for bubble columns. *AIChE journal* **53** (1), 9–18.

Monin, Andrej Sergeevič & Ozmidov, Rostislav Vsevolodovič 1985 *Turbulence in the Ocean*, , vol. 3. D Reidel Publishing Company.

Mudde, RF, Harteveld, WK & Van den Akker, HEA 2008 Uniform flow in bubble columns. *Industrial & Engineering Chemistry Research* **48** (1), 148–158.

Nam, Keeyeol, Ott, Edward, Antonsen Jr, Thomas M & Guzdar, Parvez N 2000 Lagrangian chaos and the effect of drag on the enstrophy cascade in two-dimensional turbulence. *Physical review letters* **84** (22), 5134.

Nastrom, GD & Gage, K So 1985 A climatology of atmospheric wavenumber spectra of wind and temperature observed by commercial aircraft. *Journal of the atmospheric sciences* **42** (9), 950–960.

Neufeld, Zoltán, López, Cristóbal, Hernández-García, Emilio & Tél, Tamás 2000 Multifractal structure of chaotically advected chemical fields. *Physical Review E* **61** (4), 3857.

Nickolls, John, Buck, Ian, Garland, Michael & Skadron, Kevin 2008 Scalable parallel programming with cuda. *Queue* **6** (2), 40–53.

Obukhov, AM 1941 On the distribution of energy in the spectrum of turbulent flow. In *Dokl. Akad. Nauk SSSR*, , vol. 32, pp. 22–24.

Okubo, Akira 1970 Horizontal dispersion of floatable particles in the vicinity of velocity singularities such as convergences. In *Deep sea research and oceanographic abstracts*, , vol. 17, pp. 445–454. Elsevier.

Opper, Manfred & Saad, David 2001 *Advanced mean field methods: Theory and practice*. MIT press.

Orszag, Steven A 1969 Numerical methods for the simulation of turbulence. *Physics of Fluids* **12** (12), II–250.

Oseledec, Valery Iustinovich 1968 A multiplicative ergodic theorem. lyapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc* **19** (2), 197–231.

Ott, Edward 2002 *Chaos in dynamical systems*. Cambridge university press.

Paret, Jérôme, Jullien, Marie-Caroline & Tabeling, Patrick 1999 Vorticity statistics in the two-dimensional enstrophy cascade. *Physical review letters* **83** (17), 3418.

Paret, Jérôme & Tabeling, Patrick 1997 Experimental observation of the two-dimensional inverse energy cascade. *Physical review letters* **79** (21), 4162.

Paret, Jerome & Tabeling, Patrick 1998 Intermittency in the two-dimensional inverse cascade of energy: Experimental observations. *Physics of Fluids (1994-present)* **10** (12), 3126–3136.

Pope, Stephen B 2000 *Turbulent flows*. Cambridge University Press.

Pouquet, Annick, Sen, A, Rosenberg, D, Mininni, Pablo Daniel & Baerenzung, J 2013 Inverse cascades in turbulence and the case of rotating flows. *Physica Scripta* **2013** (T155), 014032.

Pumir, Alain, Shraiman, Boris I & Siggia, Eric D 1991 Exponential tails and random advection. *Physical review letters* **66** (23), 2984.

Renyi, Alfred 1970 Probability theory. 1970. *North-Holland Ser Appl Math Mech* .

Richardson, Lewis F 1926 Atmospheric diffusion shown on a distance-neighbour graph. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **110** (756), 709–737.

Richardson, Lewis Fry 2007 *Weather prediction by numerical process*. Cambridge University Press.

Rivera, Michael & Wu, Xiao-Lun 2000 External dissipation in driven two-dimensional turbulence. *Physical review letters* **85** (5), 976.

Rose, HA & Sulem, PL 1978 Fully developed turbulence and statistical mechanics. *Journal de Physique* **39** (5), 441–484.

Rutgers, Maarten A 1998 Forced 2d turbulence: experimental evidence of simultaneous inverse energy and forward enstrophy cascades. *Physical review letters* **81** (11), 2244.

Salmon, Rick 1998 *Lectures on geophysical fluid dynamics*. Oxford University Press.

Scatamacchia, R, Biferale, L & Toschi, F 2012 Extreme events in the dispersions of two neighboring particles under the influence of fluid turbulence. *Physical review letters* **109** (14), 144501.

Schäfer, K, Thomas, W, Peters, A, Ries, L, Obleitner, F, Schnelle-Kreis, J, Birmili, W, Diemer, J, Fricke, W, Junkermann, W *et al.* 2011 Influences of the 2010 eyjafjallajökull volcanic plume on air quality in the northern alpine region. *Atmos. Chem. Phys* **11** (16), 8555–8575.

Seychelles, Fanny, Ingremeau, François, Pradère, Christophe & Kellay, Hamid 2010 From intermittent to nonintermittent behavior in two dimensional thermal convection in a soap bubble. *Physical review letters* **105** (26), 264502.

Shi, Ji Ping, Evans, Douglas E, Khan, AA & Harrison, Roy M 2001 Sources and concentration of nanoparticles ($<$ 10nm diameter) in the urban atmosphere. *Atmospheric Environment* **35** (7), 1193–1202.

Shotorban, Babak & Balachandar, S 2006 Particle concentration in homogeneous shear turbulence simulated via lagrangian and equilibrium eulerian approaches. *Physics of Fluids (1994-present)* **18** (6), 065105.

Shraiman, Boris I & Siggia, Eric D 2000 Scalar turbulence. *Nature* **405** (6787), 639–646.

Simonin, O, Deutsch, E & Minier, JP 1993 Eulerian prediction of the fluid/particle correlated motion in turbulent two-phase flows. In *Advances in Turbulence IV*, pp. 275–283. Springer.

Smith, Leslie M, Chasnov, Jeffrey R & Waleffe, Fabian 1996 Crossover from two-to three-dimensional turbulence. *Physical review letters* **77** (12), 2467.

Smithr, Leslie M & Yakhot, Victor 1994 Finite-size effects in forced two-dimensional turbulence. *Journal of Fluid Mechanics* **274**, 115–138.

Sozza, A, Boffetta, G, Muratore-Ginanneschi, P & Musacchio, Stefano 2015 Dimensional transition of energy cascades in stably stratified forced thin fluid layers. *Physics of Fluids (1994-present)* **27** (3), 035112.

Sreenivasan, Katepalli R & Antonia, RA 1997 The phenomenology of small-scale turbulence. *Annual review of fluid mechanics* **29** (1), 435–472.

Succi, Sauro 2001 *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press.

Sundaram, Shivshankar & Collins, Lance R 1997 Collision statistics in an isotropic particle-laden turbulent suspension. part 1. direct numerical simulations. *Journal of Fluid Mechanics* **335**, 75–109.

Takens, Floris 1985 On the numerical determination of the dimension of an attractor. In *Dynamical systems and bifurcations*, pp. 99–106. Springer.

Tambour, Y 1980 A sectional model for evaporation and combustion of sprays of liquid fuels. *Israel Journal of Technology* **18** (1-2), 47–56.

Taylor, Geoffrey I 1921 Diffusion by continuous movements. *Proceedings of the london mathematical society* **20**, 196–211.

Tenneti, S, Garg, R, Hrenya, CM, Fox, RO & Subramaniam, S 2010 Direct numerical simulation of gas–solid suspensions at moderate reynolds number: quantifying the coupling between hydrodynamic forces and particle velocity fluctuations. *Powder Technology* **203** (1), 57–69.

Teyssier, Romain 2002 Cosmological hydrodynamics with adaptive mesh refinement-a new high resolution code called ramses. *Astronomy & Astrophysics* **385** (1), 337–364.

Thalabard, Simon, Krstulovic, Giorgio & Bec, Jérémie 2014 Turbulent pair dispersion as a continuous-time random walk. *Journal of Fluid Mechanics* **755**, R4.

Theiler, James 1988 Lacunarity in a best estimator of fractal dimension. *Physics Letters A* **133** (4), 195–200.

Trusilova, K, Rödenbeck, C, Gerbig, C & Heimann, M 2010 Technical note: A new coupled system for global-to-regional downscaling of co 2 concentration estimation. *Atmospheric Chemistry and Physics* **10** (7), 3205–3213.

Tsang, Yue-Kin, Ott, Edward, Antonsen Jr, Thomas M & Guzdar, Parvez N 2005 Intermittency in two-dimensional turbulence with drag. *Physical Review E* **71** (6), 066313.

Tulloch, R & Smith, KS 2006 A theory for the atmospheric energy spectrum: Depth-limited temperature anomalies at the tropopause. *Proceedings of the National Academy of Sciences* **103** (40), 14690–14694.

Van Kampen, Nicolaas Godfried 1992 *Stochastic processes in physics and chemistry*, , vol. 1. Elsevier.

Walker, James S 1996 *Fast fourier transforms*, , vol. 24. CRC press.

Warhaft, Z 2000 Passive scalars in turbulent flows. *Annual Review of Fluid Mechanics* **32** (1), 203–240.

Weiss, John 1991 The dynamics of enstrophy transfer in two-dimensional hydrodynamics. *Physica D: Nonlinear Phenomena* **48** (2-3), 273–294.

White, Christopher M & Mungal, M Godfrey 2008 Mechanics and prediction of turbulent drag reduction with polymer additives. *Annu. Rev. Fluid Mech.* **40**, 235–256.

Wilkinson, M & Mehlig, Bernhard 2003 Path coalescence transition and its applications. *Physical Review E* **68** (4), 040101.

Williams, FA 1958 Spray combustion and atomization. *Physics of Fluids (1958-1988)* **1** (6), 541–545.

Wotawa, G, De Geer, L-E, Becker, A, D'Amours, R, Jean, M, Servranckx, R & Ungar, K 2006 Inter-and intra-continental transport of radioactive cesium released by boreal forest fires. *Geophysical research letters* **33** (12).

Xiao, Z, Wan, M, Chen, S & Eyink, GL 2009 Physical mechanism of the inverse energy cascade of two-dimensional turbulence: a numerical investigation. *Journal of Fluid Mechanics* **619**, 1–44.

Yang, Shu-Qing 2009 Drag reduction in turbulent flow with polymer additives. *Journal of Fluids Engineering* **131** (5), 051301.

Yeung, PK, Zhai, XM & Sreenivasan, Katepalli R 2015 Extreme events in computational turbulence. *Proceedings of the National Academy of Sciences* **112** (41), 12633–12638.

Yin, Xiaolong & Sundaresan, Sankaran 2009 Fluid-particle drag in low-reynolds-number polydisperse gas–solid suspensions. *AIChE journal* **55** (6), 1352–1368.

Yu, Hongbin, Chin, Mian, Yuan, Tianle, Bian, Huisheng, Remer, Lorraine A, Prospero, Joseph M, Omar, Ali, Winker, David, Yang, Yuekui, Zhang, Yan *et al.* 2015 The fertilizing role of african dust in the amazon rainforest: A first multi-year assessment based on data from cloud-aerosol lidar and infrared pathfinder satellite observations. *Geophysical Research Letters* **42** (6), 1984–1991.

Zaichik, LI, Simonin, O & Alipchenkov, VM 2009 An eulerian approach for large eddy simulation of particle transport in turbulent flows. *Journal of Turbulence* (10), N4.

Acknowledgments

Working in order to gain a PhD diploma is something I have found really, really interesting. At the end of this journey, I personally see a three groups classification in the set of skills a PhD candidate has to develop during his thesis: scientific, technical, and managing.

First one encompasses the core of your work. Gathering present-day knowledge about a specific topic. Designing experiments to be performed. Presenting scientific work and results through talks, posters or articles to a variety of audiences: students, peers, scientific from other communities, private investors... There is a shining excitement, and a subtle beauty in all this process.

Second group may vary a lot, and is related to the tools you have actually used for your work: building and using variety of top tech machines. I didn't have the opportunity to do what we call "experimental work", but from what I know, the issue is the same as in a "numerical work": devil is in the details. You have to be careful about everything, every setup, every screw, every line of code, every possible bug... you have to make sure that the machine that you use indeed does what you want it to do ! Otherwise your results might be irrelevant and you have to start all over again.

The third is the human skills that you acquire. Depending on if you had to manage people or not: how to give motivation but mostly how to keep self-motivation on long-lasting projects. I personally did learn a lot from those tasks and I think that this set of skills is what give PhDs a great autonomy and project-based way of thinking and working.

I did also spent a lot of time building what I called technical skills. I apologize for that to my close scientific co-workers, but I know that they know that it is unfortunately an increasing need in nowadays competitive society, more oriented toward short-terms profits rather than fundamental research.

What makes the PhD journey even more interesting is that each is shaped differently depending on its environment. Did the candidate get all the necessary funding to successfully carry his / her research, to benefit from an adequate continuous formation, or was he concerned about tight budget management ? Did he had to work mainly alone ? Did he had to manage interns or engineers ? Did he had to, or choose to focus on technical, engineering details or requirements from his lab or investors ?

Without doubt, I was lucky enough to benefit from one very enviable working environment. A (too) nice place to work in, with the Nice Observatory being situated in the heights of Nice and my living place in the very charming town of Menton, surely I was living in a dreamy sunny place. As a mountain lover that I was during my PhD, this allowed me to run countless of kilometres and discover truly amazing places. Sport is something that had a central importance aside of my work charge, allowing a good equilibrium between intellectual and physical loads.

Very nice places to break the day: many thanks to the restaurant staff of the observatory: Karima, Nadia, Khaled, Ghislain... A good restaurant in a company makes possible fruitful discussions in an enjoyable place and they for sure did a wonderful job throughout those years. The little kitchen in which I've spent countless of coffee breaks was also a nice place to bring the needed serenity between work sprints.

Nice colleagues and friends: I was very pleased to work with the others PhD students and post-docs I've met during my thesis: Mamadou, Christophe S, Christophe P and Christophe H, Judit, Sophia, Simon... And Giorgio Krstulovic as co-advisor, who also initiated me to the art of rock climbing.

An amazing thesis director, displaying a rare combination of human, management and scientific skills. Giving responsibility without ceasing to believe in his people in charge. Always patient and ready to help. For all this, he is now one of my primary source of inspiration for my upcoming career. I've met some (hopefully not that many) PhD candidates experiencing disastrous professional relationships with their thesis director, and this is truly sad, considering how this factor is one of the most important and determining factor during a PhD thesis.

I wish to thank all the administrative staff from the observatory, Nice university and CNRS, especially Rose Pinto, Elizabeth Taffin-de-Givenchy, Monique Clatot, for their help in all administrative stuffs, making them less ever-lasting. A big thank you to many people with which I had very interesting technical and / or scientific dicussions: Fabrice Ubaldi, Uriel and Helene Frisch, Annick Pouquet, Pablo Mininni, Berangere Dubrulle, Guido Boffetta, Stefano Musacchio, Rahul Pandit, Samriddhi Sankar Ray... while I don't want to miss anyone, I could not cite them all here. I've learned a lot from those people who brought me motivation and inspiration.

I was also lucky to work and discuss with people from which I did learn a lot about science politics and higher education in France. Those include people from the team of the association for the young researchers of the Maritime Alps, people from the scientific council of the Nice university, and once again my thesis advisor.

Thank you to my friends and collaborators in Belgium: Julien, Charles, Michel, my sister Marie, from which I was able to stay up to date about business and human management.

Thank you to my family and my beloved Celine. It is never evident for non-researchers to understand the relevance and implication of the work that a PhD candidate has to carry, neither the irregular working loads, countless late evenings that it requires. They were nevertheless very comprehensible, bringing their own personal colouring to my work, and for some of them, always ready for party.

The combination of all these factors and people I've met these years have resulted in a fantastical experience during which time flew way too fast, and that shaped my brain and my heart in a way I couldn't have hoped better.