Università
degli Studi
di Padova

University of Padova

Department of Industrial Engineering

Master Thesis in Aerospace Engineering

Development and implementation of a stochastic model for the simulation of inertial particles in turbulent flows

Supervisor
Prof. Francesco Picano

Master Candidate
Crescenzi Alessandro,
Matr. 2021173

Co-supervisor
Prof. Federico Dalla Barba

October 20, 2023

Ai miei nonni,
anche se l'inglese non lo sanno leggere

Abstract (IT)

Nella simulazione di flussi turbolenti, le Wall-Modeled Large-Eddy Simulations (WMLES) sono una soluzione eccellente grazie al loro equilibrio tra accuratezza e costo computazionale, così come le RANS supportate da modelli specifici. Tuttavia, sia le RANS che le WMLES mostrano alcune difficoltà nell'emulare il comportamento del flusso in prossimità delle pareti, a causa della risoluzione non sufficientemente elevata per risolvere direttamente l'intero strato limite. Questo problema si aggrava quando si introduce il trasporto di particelle inerziali, perché in una turbolenza wall-bounded, a parete i corpi sono particolarmente influenzati dalle forze di taglio e dall'anisotropia.

In questo contesto, il presente lavoro mira a implementare un modello stocastico per il trasporto di particelle inerziali in flussi turbolenti non risolti. Partendo da un flusso turbolento medio dato da un modello, il comportamento delle particelle è derivato da un modello che implementa un'equazione differenziale stocastica basata sul Continuous Random Walk (CRW) normalizzato di Langevin, che simula le fluttuazioni del fluido viste dalle particelle utilizzando un modello ibrido lagrangiano-euleriano, aggirando completamente il problema della risoluzione. In questo lavoro di tesi, l'attenzione si concentra sul comportamento del flusso turbolento carico di particelle in un canale piatto e sul modo in cui le particelle si depositano sulle pareti. I risultati di questo modello saranno poi confrontati con lavori precedenti e con i dati di Direct Numerical Simulation(DNS) legati ai parametri iniziali in cui è stato sviluppato il modello, cioè a Re_τ=150. Una volta dimostrata la validità del modello, questo verrà applicato a contesti più vicini alla realtà, aumentando il numero di Reynolds, per valutare se il modello è ancora efficace in casi più complessi.

Abstract (EN)

When simulating turbulent flows, Wall-Modeled Large-Eddy Simulations (WMLES) are an excellent solution due to their balance between accuracy and computational cost, as well as RANS aided from proper models. However, RANS and Wall-Modeled LES shows some difficulties in emulating the flow behaviour near the walls, due to the resolution being not high enough to directly resolve the entire boundary layer. This problem is aggravated when the transport of inertial particles is introduced, because in a wall-bounded turbulence the bodies are particularly affected by the shear and the anisotropy near the walls.

In this context, the present work aims at implementing a stochastic model for the transport of inertial particles in unresolved turbulent flows. Starting from a mean turbulent flow given by a model, the particle behaviour is derived by a model that implements a stochastic differential equation based on the normalised Langevin continuous random walk (CRW), which simulates the fluid fluctuations seen by the particles using a hybrid Lagrangian-Eulerian model, completely bypassing the resolution problem. In this framework the focus is on how this particle-laden turbulent flow behaves in a flat channel, and on how the particles deposit on the walls. The results of this model will then be compared with previous work and also Direct Numerical Simulation (DNS) data, linked to the initial parameters in which the model was developed, i.e. at Re_τ=150. Once the consistency of the model has been demonstrated, the model will be applied to contexts closer to reality, i.e.increasing the Reynolds number, to assess whether the model still holds in more complex cases.

Contents

Listing of figures

Listing of tables

Listing of acronyms

ACS	Attitude Control System
b	Width of the channel
β	Correlation factor between T_L and T_E
CRW	Continuos Random Walk
C_S	Smagorinksi Constant
C_W	WALE Model Smagorinski Constant
δ	Half height of the channel
δ_ν	Viscous lengthscale
DNS	Direct Numerical Simulation
d_p	Particle Diameter
DRW	Discrete Random Walk Model
E	Strain-rate Tensor
ϵ	Fluid Energy Dissipation Rate
η	Kolmogorov Scale
γ	Stretching Factor
$G_{\triangle}$	LES filtering operator
h	channel height
LES	Large Eddies Simulation
μ	Dinamic viscosity
ν	Cinematic viscosity
ν_r	Residual viscosity
ν_t	Turbulent viscosity
p	Local Pressure
p_w	Wall Pressure

RANS	Reynolds-Averaged Navier-Stokes
Re	Reynolds number
Re_η	Reynolds number at the Kolmogorov Scale
Re_p	Particle Reynolds number
Re_τ	Friction Reynolds number
ρ	Density
ρ_f	Fluid Density
ρ_p	Particle Density
$\overline{S}_{ij}$	Residual Strain-rate tensor
σ_i	Fluid velocity deviations
Stk	Stokes number
$\mathbf{T}$	Viscous stress tensor
τ_L	Fluid relaxation time
τ_η	Characteristic time at Kolmogorov Scale
τ_p	Particle relaxation time
τ_R	Reynolds stress tensor
$\tilde{\tau}_R$	Residual stress tensor
τ_w	Wall shear stress
T_E	Fluid Eulerian integral time scale
TKE	Turbulent Kinetic Energy
T_L	Fluid Lagrangian integral time scale
U_b	bulk velocity
u_τ	Friction velocity
WALE	Wall-Adaptive Large Eddy
y^+	Wall coordinate

1

Introduction

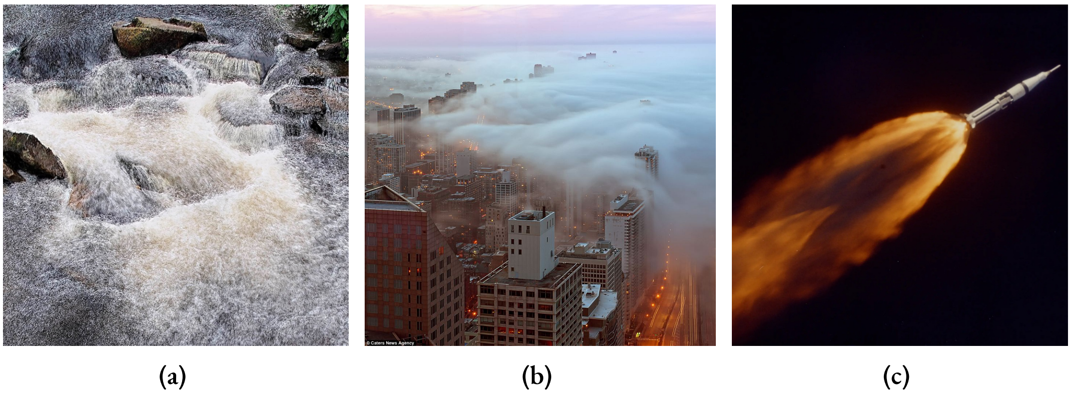
Something that is often ignored in daily life is that gas itself is a fluid, like water or any other liquid.

In this light, it is clear how much fluids are present anywhere you look.

These two categories are grouped together when explaining their physics, due to their similar behaviour.

Their study is vast and wide-ranging, spanning various fields of science and engineering, aspects of everyday life and beyond. For fluid dynamics, the study of fluids motion, some of these aspects are more complicated to study and reproduce, either experimentally or artificially.

One of them is the subject of this thesis: turbulence is one of the most common phenomena in fluids, characterised by the interaction of many vortex structures of different sizes, resulting in chaotic and multiscale processes.



(a) (b) (c)

Figure 1.1: Some turbulence examples.

Ranging from mostly natural domains, such as the swirls in a flowing river, the puffs of a volcano or even the movement of weather disturbances, to more man-dependent phenomena, like the flow of wind between buildings in a city, turbulence is virtually omnipresent.
Even in the engineering field turbulence is relevant: the plumes from a rocket motor, the flow of the air after it has passed the wing of aircraft, and so on. Many of these phenomena are the subject of very active field of research, characterised by analysis done using numerical simulation.

For this kind of problem, Large Eddies Simulation (LES) is the most widely used numerical tool today. In a nutshell, LES directly resolves the largest scales of the turbulent motion on the computational grid, enabling the most crucial phenomena to be captured.
On the downside, all the smallest scales of the motion cannot be simulated and are are therefore modeled via ad-hoc closure models, reducing the computational cost and the required grid solution.
Another problem, more relevant in the context of this work, is the numerical treatment of the boundary layer.
There are two ways to address this issue: Wall-Resolved LES and Wall-Modeled LES. The former uses a grid resolution small enough to resolve the boundary layer up to the wall. This allows much more reliable simulations, but at the cost of being computationally very expensive,making it implausible at high Reynolds numbers even with today's computer capabilities.

Instead, the wall-modeled LES employs a wall model to describe the velocity profile within the boundary layer. This way a less dense grid is required, making high Reynolds simulations feasible.

However, what characterises the work carried out in this thesis is something else.
In many of the situations already described, there is a ubiquitous agent that has not yet been considered: particles.
Unless a sterile and completely sealed environment is established, particles will inevitably be present in the fluid under examination. Despite their seemingly minimal significance, there exists a phenomenon called turbophoresis that certifies the importance of their presence.
Turbophoresis refers to the migration of a suspended particle in a fluid towards a decreasing turbulence level. This accumulation of particles can lead to a "remodelling" of the wall shape. in fact, particles tend to accumulate in the wall region and can adhere to the wall, resulting in wall shape remodeling.
Undoubtedly, such a phenomenon can be very important in many engineering applications, especially in the aerospace sector: the behaviour of the particles in the exhaust of a rocket engine in relation to the nozzle wall; the presence of particulate matter in the ACS of any satellite's propellant; the suspended dust in a martian devil dust.
Such phenomena cannot be underestimated, which is why efforts are now being made to extend the simulation library. Especially in the modeling of suspended particles in a turbulent channel flow, which is the most simple and adaptable case study, and therefore is the one that will be analyzed in this work .

Simulating the presence of particles and turbulent phenomena in any kind of simulation proves challenging as the models developed may not be valid for all turbulence values (which can be represented by a variable, the Reynolds number, explained later). The Reynolds number closely correlates to simulation complexity, and turbulent flows have high Reynolds numbers, which causes simulation difficulties. As turbulence increases, simulation issues become more and more pronounced.

However, frequently the models utilized to aid with LES or RANS are created at low Reynolds numbers, which simplifies its development, but this does not ensure their efficacy when implemented in practical conditions.
The purpose of this thesis is to determine a model that can assist in both LES and RANS and evaluate its accuracy by comparing its outcomes with those present in the existing literature. The aim is also to extend the model to higher Reynolds values to assess whether the selected model is still operational. The Continuos Random Walk(CRW) model based on the normalized Langevin equation was selected. The use of the edition used here started with Marchioli's work in 2006 and has continued since then to validate its effectiveness.

This dissertation will be structured as follows: Chapter 2 provides a brief introduction to all the subjects studied at a physical level; Chapter 3 offers an overview of fluid-dynamic simulations, both from a fluidic and particle perspective; Chapter 4 focuses on the implementation and verification of the model compared to the original work; Chapter 5 will extend the model to a higher Reynolds number, and evaluate the obtained results; Chapter 6 will conclude the thesis with a brief analysis on the effectiveness and goodness of the model, and outline potential future advancements.

2
Background

To better understand the kind of work conducted in this thesis, a brief review of its fundamental physics will be provided: Section 2.1 will focus on the treatment of turbulence, in Section 2.2 motion scales will be examined, while Section 2.3 analyses wall-bounded flows behaviour, finally in Section 2.4 the characteristics of particles and turbophoresis are explored.

2.1 Turbulence

Turbulence is a widespread phenomenon: from the water in a waterfall, to the dust in a strong wind, to the exhaust plume from a solid rocket motor and many more, turbulent flows can be observed all around us.

Something it is possible to catch on by these examples is that the flow exhibits an unpredictable nature, which is evident from the presence of vortex structures of different sizes, and the overall chaotic nature of the flow, which appears unsteady and irregular, both in position and time.

These features render this flow type significantly more efficient in mass transportation and mixing capacity than laminar flows.

This and the presence of turbulence in numerous engineering applications deems it a important study field. Unfortunately, studying this subject is not an easy feature, given that its mathematical models are so complex that sometimes they introduce new problems in the whole system equation [20].

That's due to the fact that analytical solutions to fluid dynamics equations are typically straightforward only in the laminar flow regime; in a turbulent regimes, there's no such thing, as experimentation and/or numerical models need to be applied. That turns the whole process in a way more complex and time-consuming matter.

Thus, it is essential to define a parameter that allows to differentiate between the two scenarios, and this can be achieved through the Reynolds number (*Re*), defined as:

$$Re = \frac{uL}{\nu} = \frac{\rho u L}{\mu} \tag{2.1}$$

This definition of *Re* describes it as the ratio between inertial and viscous forces: in turbulent flows, the former dominates over the latter, while in laminar flows the opposite is true. However, there is no exact value of *Re* at which this distinction is made, even though a generic separation value is put at Re=1500.

When approaching turbulence, the deterministic chaos problem comes into play, as small variations in initial conditions can lead to significantly different outcomes. This sensitivity of the system to small fluctuations can cause significant differences in the final solution.

Still, when considering the average, the outcome will be unchanged.

Therefore, even though the mean solution is not affected by deterministic chaos, the instantaneous one is, thus requiring a statistical approach. To achieve this, a large number of instantaneous fields N need to be analysed, upon which a Reynolds or ensemble average is applied:

$$< U^n(t) >= \frac{1}{N}\sum_{1}^{N} u^n(t) =< U > \tag{2.2}$$

Here the vector U represents the mean velocity field. When examining any velocity field, such as for example U(x,t), it is always possible to utilise a Reynolds decomposition:

$$U(x,t) =< U(x,t) + u'(x,t) > \tag{2.3}$$

where the first component is the above mentioned mean velocity field, and the other one is the fluctuation component.

From this operation a velocity profile like the one represented in Figure 2.1 is obtained.

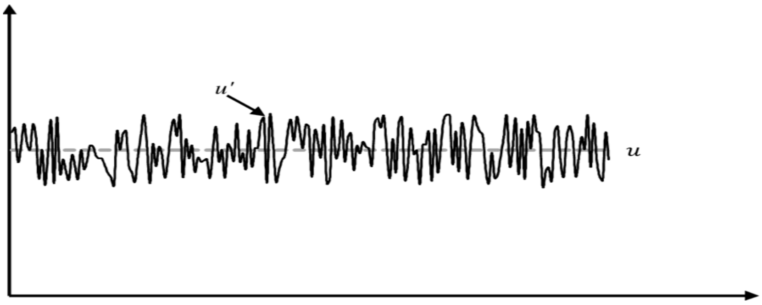


Figure 2.1: Velocity fluctuation obtained through Reynolds decomposition [15].

2.2 Kolmogorov Scales

A turbulent flow will always display 3D non-stationary vortex structures spanning a great range of size orders: this means that in structures like clouds, vortexes of sub-millimetre dimensions exist along with kilo-metre ones, meaning that there can be a six orders of magnitude difference.

And, as the Reynolds number increases, the dimension of the vortex achievable gets smaller and smaller.

To understand the behaviour of turbulent flows, an explanation by Lewis Fry Richardson can be used: when observing a fountain jet flowing into a pool, it is possible to observe various turbulent formations propagating on the surface of the water; while the behaviour of large scales is affected by the system's geometry, smaller scales remain unaffected by it, and

they can always complete their motion cycle without encountering physical obstacles; therefore they have an universal behaviour; still, smaller scales formations seem to stem from bigger ones.
This is well synthesised in a poem written written by the scientist:

Big whorls, little whorls

Big whorls have little whorls
that feed on their velocity,
And little whorls have lesser whorls
and so on to viscosity.

Lewis Fry Richardson (1922)

Richardson introduces the concept of energy cascade to explain these behaviours objectively. The first concept he introduced is that the turbulent flow is made up of eddies: there is no precise definition, but an eddy is conceived as a turbulent motion within a certain area, that is at least moderately coherent over that region. Additionally, a single large eddy's region can comprise several smaller eddies.
Eddies can be of any size: they will have a characteristic dimension l, a characteristic velocity $u(l)$ and a characteristic time $\tau(l) = l/u(l)$.
In addition, an homogeneity and isotropic property is verified for any eddy.

According to Richardson's theory, the scales that are the largest possess the maximum amount of kinetic energy. These bigger eddies are unstable, and as they break up the energy will gradually transfer to smaller and smaller eddies. At the end of the cycle, the energy is dissipated in the smallest eddies by viscosity agents, resulting in the generation of the energy cascade. [21].
This happens because at larger scales, where $l \sim l_0 \sim L_0$, L_0 being the system dimension, both the eddy viscosity and the eddy velocity will be comparable to the system ones.
By defining $Re_l = \frac{u_{l_0} l_0}{\nu}$, which in this case will be large, it is highlighted that the direct effects of viscosity are negligible, while inertial ones are dominant, therefore no energy is lost there.

Instead the smaller scales are such that $l << L_0$, thus the dimensional analysis shows that $Re_\eta \sim 1$, meaning that the viscous terms there are relevant, resulting in energy dissipation. This explains why the eddies energy is simply transferred in every single larger scale, remaining constant until a certain small scale is reached.
The scale where dissipation occurs is called Kolmogorv Scale η, defined as the scale small enough to make the eddy motion stable in the turbulent flow. At that point the energy cascade ends, as eddies can not get much smaller before dissolving.
A proper schematic representation of all the scales and dimensions described is given in Figure 2.2

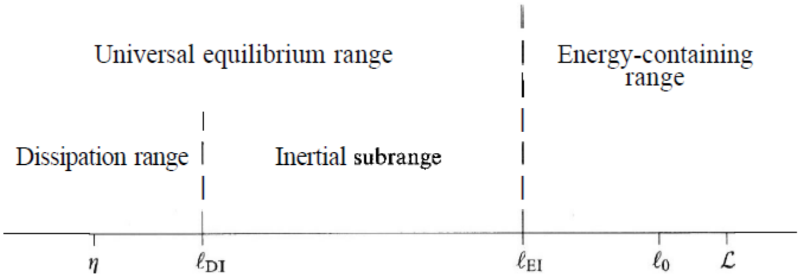


Figure 2.2: Eddies lenghtscales and ranges in Richardson's theory[20].

The concept expressed by Richardson needs some clearer definitions, especially from a dimensional point of view. This is comprehensively done by Kolmogorov's K41 theory. The Russian mathematician summarizes the fundamental principles in three hypothesis[20]:

- **Hypothesis "0"**, or **Local isotrophy hypothesis**: when $Re >> 1$ e $l << l_0$, the fluid is locally homogeneous and isotropic, thus the properties will be universal and at most depend on ϵ, ν

- **First similarity hypothesis**: every single statistics of the small-scale motion ($l < l_\eta$) has a universal form that is uniquely determined by ν and ϵ, in every turbulent flow at sufficiently high Reynolds number

- **Second similarity hypothesis**: in every turbulent flow with $Re \gg 1$, the statistics of the eddies of scale l in the range $\eta << l << l_0$ are universal and dependant only on ϵ not on ν

In the presented hypothesis ν is the fluid viscosity, ϵ is the fluid energy dissipation rate. Using the energy dissipation rate it is possible to define the characteristic velocity scales and timescales of an eddy, given its dimension l:

$$u(l) = (\epsilon l)^{1/3} = u_\eta \, (l/\eta)^{1/3} \sim u_0 \, (l/l_0)^{1/3}$$

$$\tau(l) = (l^2/\epsilon)^{1/3} = \tau_\eta \, (l/\eta)^{2/3} \sim \tau_0 \, (l/l_0)^{2/3}$$

Also, by doing a dimensional analysis of the rate at which the energy is transferred in the inertial sub-range, it can be assumed to be equal to the energy dissipation rate ϵ.

It is now evident that smaller vortex structures are unaffected by geometry or perspective, and that is attributable to their isotropic property.

Smaller scales exhibit an universal behaviour, determined by the transfer of kinetic energy and viscosity. There will be a range, smaller than l_0 and larger than η where the energy is transferred from vortex to vortex; the upper limit represents the point where all the energy is stored, while the lower limit represents the point at which the energy is dissipated.

A relation between these two scales can be obtained through dimensional analysis:

$$\frac{l_0}{\eta} \propto Re_0^{3/4} \qquad \frac{u_0}{u_\eta} \propto Re_0^{1/4} \qquad \frac{\tau_0}{\tau_\eta} \propto Re_0^{1/2}$$

From these relations, in this study case of interest, which means turbulent flow and higher *Re*, this will provide an increasingly larger inertial range, while time scales and velocities decrease gradually: from a computational point of view, that is completely detrimental since it requires more of every resource.

2.3 Wall-Bounded Flows

In most nature and engineering applications, turbulent flows are usually bounded between a couple of solid surfaces at least. Therefore the focus of this thesis will be on a channel flow, and for this reason a brief review on the subject is required.

The classical theory case, which is pretty similar to the one analyzed later, will be described: the channel has rectangular section of heigth $h = 2\delta$, length $L/\delta >> 1$ and width $b/\delta >> 1$, as depicted in Figure 2.3.
The flow main direction follows the X-axis, the mean cross-stream velocity is equal to zero.

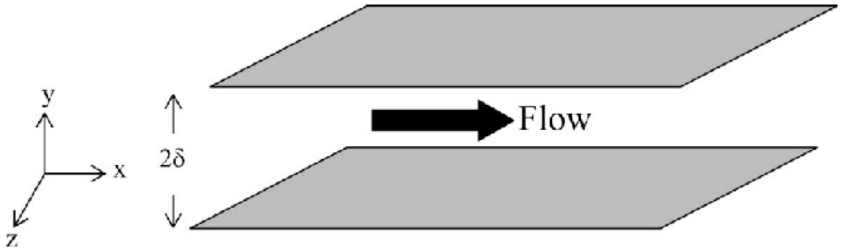


Figure 2.3: Sketch of the channel flow [9].

Also, once the flow is fully developed, the channel enters a full-development region, where velocity statistics will not change in the X direction.
Hence the fully developed channel flow is considered statistically stationary and one-dimensional (depending only on the Y coordinate).
The Reynolds number in the channel is defined as:

$$Re = \frac{2\delta U_b}{\nu} \tag{2.4}$$

where U_b is the bulk velocity, expressed as:

$$U_b = \frac{1}{\delta}\int_0^{\delta} \langle U \rangle dy \tag{2.5}$$

Regarding the balance of mean forces, starting from Reynolds-Averaged Navier-Stokes (RANS):

- the mean continuity equation reduces to:

$$\frac{\partial \langle V \rangle}{\partial y} = 0 \tag{2.6}$$

- the lateral mean-momentum equation reduces to:

$$\frac{\partial}{\partial y}(p + \rho\langle v'^2\rangle) = 0 \tag{2.7}$$

- the axial mean-momentum reduces to:

$$-\frac{p_w}{dx} + \mu\frac{\partial^2 U}{\partial y\,\partial y} - \rho\frac{\partial}{\partial y}\langle u'v'\rangle = 0 \tag{2.8}$$

When the boundary condition $\langle v^2\rangle_{y=0} = 0$ is applied, from Equation (2.7) derives that the mean axial pressure gradient is uniform across the flow, giving back $\frac{\partial p}{\partial x} = \frac{\mathrm{d}P_w}{\mathrm{d}x}$ for every y. Instead, Equation (2.8) can be rewritten as:

$$\frac{\mathrm{d}P_w}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}y}\left(\mu\frac{\mathrm{d}u}{\mathrm{d}y} - \rho\langle u'v'\rangle)\right) \tag{2.9}$$

and therefore:

$$\frac{\mathrm{d}\tau}{\mathrm{d}y} = \frac{\mathrm{d}p_w}{\mathrm{d}x} \tag{2.10}$$

Where the term $\tau(y)$ is the total shear stress, given by the sum of the viscous stresses and Reynolds stresses:

$$\tau(y) = \mu\frac{\mathrm{d}\langle U\rangle}{\mathrm{d}y} - \rho\langle u'v'\rangle \tag{2.11}$$

In this type of flow there is no mean acceleration so the mean momentum Equation (2.10) is basically a balance between the axial normal stress gradient and the cross-stream shear-stress gradient.

Still from Equation (2.10) it is apparent that τ and p_w derivatives in the y and x directions respectively are constant, thus solutions for $\tau(y)$ and p_w/x can be written explicitly using wall shear stress $\tau_w \equiv \tau(0)$.

As stated before, $\tau(y)$ is antisymmetric about the midplane, so the solutions will be:

$$-\frac{\mathrm{d}p_w}{\mathrm{d}x} = \frac{\tau_w}{h} \tag{2.12}$$

$$\tau(y) = \tau_w(1 - \frac{y}{h}) \tag{2.13}$$

Close to the wall all Reynolds stress are equal to zero, due to the boundary condition $U(x, t) = 0$, therefore the wall shear stress is fully due to the viscous input:

$$\tau_w = \rho\nu \left(\frac{\mathrm{d}\langle U\rangle}{\mathrm{d}y}\right)_{y=0} \tag{2.14}$$

At this point it is possible to obtain profiles of the shear stresses along the height of the channel (Figure 2.4). It is clear that viscous stress are dominant at the wall, while the situations reverses in free shear flows: at high Reynolds number, the viscous contribution is negligible compared to Reynolds stresses. Speaking of, since at the wall the viscosity is important, the velocity profile is related to Re.

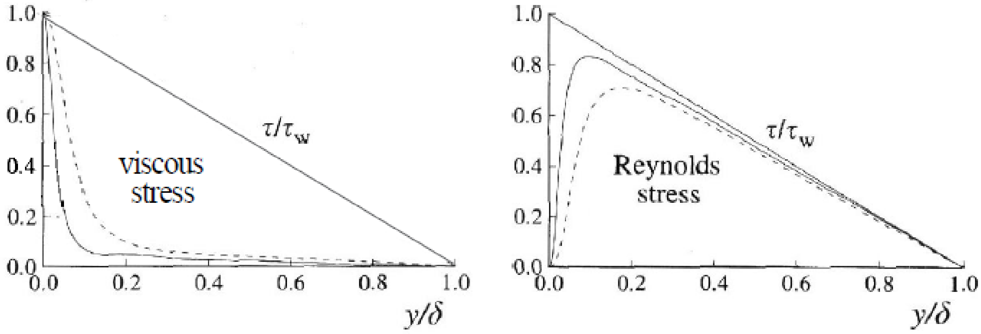


Figure 2.4: Profiles of the viscous shear stress, and the Reynolds shear stress in turbulent channel,the dotted lines represents Re=5600, the continuos one Re=13750 [20].

One other major point is that in the wall region, ν and τ_w are important parameters, as from them appropriate scales for the fluid motion in the near-wall region can be defined:

- Friction velocity:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

- Viscous lengthscale:

$$\delta_\nu = \nu\sqrt{\frac{\rho}{\tau_w}} = \frac{\nu}{u_\tau}$$

- Friction Reynolds number:

$$Re_\tau = \frac{u_\tau \delta}{\nu}$$

- Wall unit:

$$y^+ = \frac{y}{\delta_\nu} = \frac{u_\tau y}{\nu}$$

Something useful to notice is that y^+ is similar to the local Re_τ, so its magnitude can be used to determine the relative importance of viscous and turbulent processes. By plotting the two type of stresses in relation to y^+, the assertions made till now are clear: the viscous contribution drops from 100% at the wall to 10% by $y^+ = 50$ (Figure 2.5).

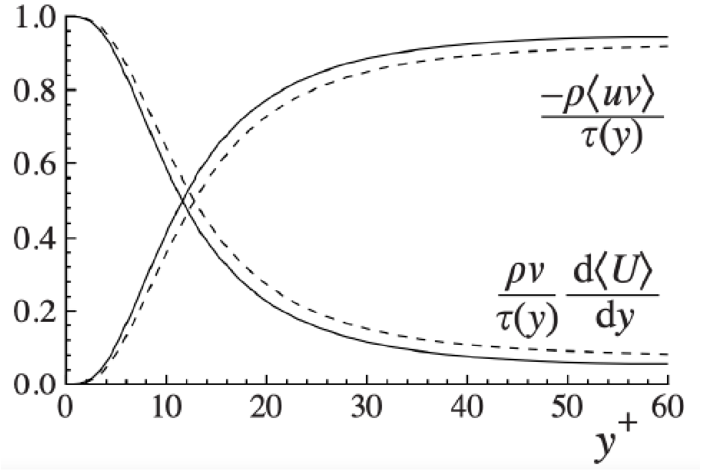


Figure 2.5: Profiles of the fractional contributions of the viscous and Reynolds stresses to the total stress. Dashed lines: Re=5600; Continuos lines:Re=13750[20].

The wall unit is also used to defined different regions in the near-wall flow, the main division being: viscous wall region $y^+ < 50$ where shear stress is influenced by the molecular viscosity; outer layer $y^+ > 50$ where they are almost irrelevant.
Due to the variety of phenomena influencing the flow as the distance from the wall increases, the laws regulating the velocity profile change multiple times (their demonstration won't be

reported): using them as starting point, another set of flow sections can be defined.
Starting from the wall: viscous sub-layer, buffer layer, log-law region, defect law. The latter are named after the trend of the velocity profile.

- In the viscous sublayer the velocity profile can be assumed equal to y^+

- In the log-law layer the velocity profile is defined by:

$$U^+ = \frac{1}{k} ln y^+ + B \tag{2.15}$$

  where B is equal to 5.2, and k is the Von Karman constant (=0.41)

- In the defect law layer the profile is defined by:

$$\frac{U_0 - \langle U \rangle}{u_\tau} = B - \frac{1}{k} ln \frac{y}{h} \tag{2.16}$$

  with B now equal to 0.3.

All of this regions can be grouped in two macro-regions, the inner layer and the outer layer, and the log-law and defect law regions together are the overlap between them.
Every region is represented in Figures 2.6 and 2.7 and then summarized in Table 2.1.

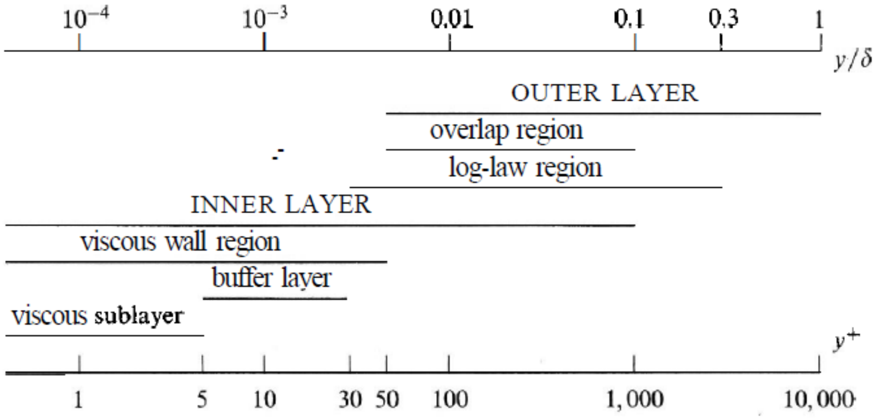


Figure 2.6: Wall regions defined in terms of y^+.

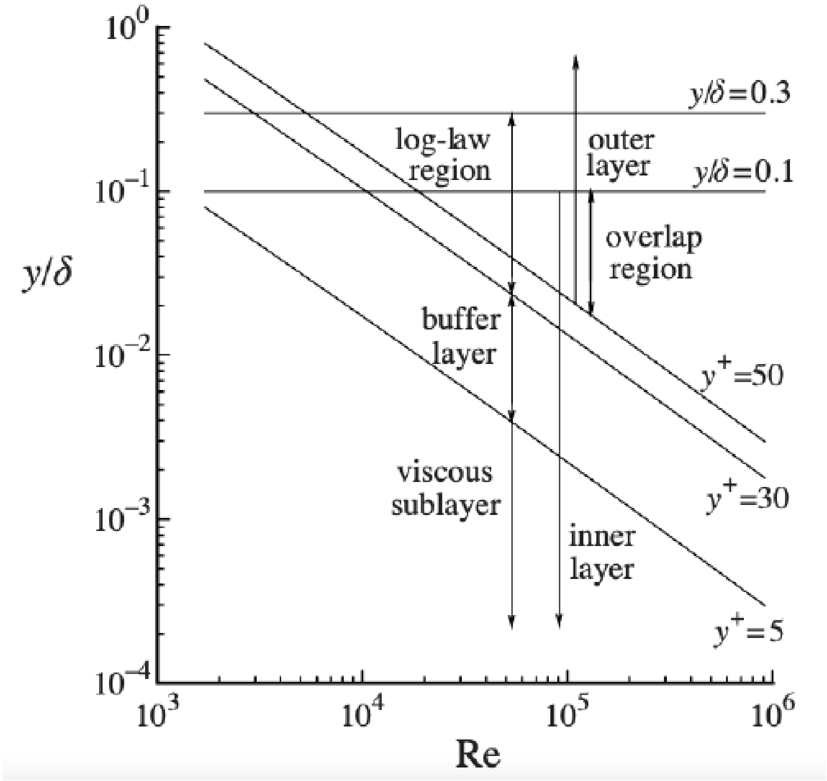


Figure 2.7: Wall regions as a function of the Reynolds number.

Region	**Location**	**Defining property**
Inner layer	$\frac{y}{\delta} < 0.1$	$\langle U \rangle$ depends only on u_tau and y^+
Outer layer	$\frac{y}{\delta_\nu} > 100$	Direct effects of viscosity are negligible
Viscous wall region	$y^+ < 50$	Viscous contribution to τ_w is significant
Viscous sublayer	$y^+ < 5$	Viscous stress totally overcome Reynolds's
Buffer layer	$5 < y^+ < 30$	Between viscous sublayer and log-law region
Log-law region	$y^+ > 30, \frac{y}{\delta} < 0.3$	Region where log-law applies
Defect-law region	$0.3 < \frac{y}{\delta} < 0.5$	Region where defect law applies
Overlap region	$y^+ > 50, \frac{y}{\delta} < 0.1$	Overlap between inner and outer layers

Table 2.1: Wall regions and layers and their defining properties.

2.4 Inertial Particles

It is now time to address the particle element of the problem. When particles are present in a stream, such flow is defined as a particle-laden flow. This refers to a type of two-phase fluid flux where one phase, known as the carrier phase, is continuously connected while the other component is made up of small, immiscible, diluted particles called the particle phase.

The modelling of these types of flows has an incredible variety of scientific applications, including aerosol and pollution in the atmosphere, dust storms, pharmaceutical processes, injection in the combustion process and many others.

To examine them, it is helpful to outline the primary physical properties added in the system by the introduction of particles into the study.

Before beginning, it is important to note that when studying particle-laden flows, there are two potential approaches available. The first approach, one-way coupling, considers only the fluid's influence on the behavior of the particles, which have no effect on the fluid's propagation. The second approach, two-way coupling, acknowledges the coexistence of fluid and particles and requires a revision of the fluid's governing equations (later presented in Section 3.1) to account for external inertial bodies' presence. When the mass fraction of the dispersed phase is sufficiently small, the assumption of one-way coupling may be considered reasonable, otherwise two-way coupling must be considered.

In the case of this study, given the computational and modelling difficulties that would otherwise arise, the first method is chosen. Now, before looking at any motion equation, an introduction to the physical quantities related to particles is necessary (using Varaksin [26] as source).

To start it is necessary to name the intensive quantities, namely the particle diameter d_p and their physical density ρ_p.

After that, a dynamic inertial parameter of particles can be introduced, defined by the time of their relaxation τ_p, obtained as:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu} \tag{2.17}$$

where μ is the dinamic viscosity, also named fluid molecular viscosity.
The relaxation time is a measure of the time it takes for a particle to approach a steady state. This time characterises the time it takes for a particle to adjust or 'relax' its velocity to a new set of forces. It indicates the particle's agility in adjusting to new environmental or conditional changes. The value is dependent on the mass and mechanical mobility of the particle and is not affected by the external forces acting on the particle.
As the size of particles increases, relaxation time grows in proportion to the square of particle diameter. Small particles typically adapt to new environments quickly by following the flow patterns, while larger particles tend to remain on their original course and be less adaptable.
If a particle is introduced into a moving airstream, it will converge to the velocity of the stream with the characteristic relaxation time τ. The characteristic time for most particles of interest to achieve steady motion in air is extremely short. A particle's velocity in a fluid rapidly adjusts to a stable state at which the drag force is in equilibrium with the other forces acting on the particle.

However, the expression presented above is a simplification of a more complicated one, that is valid only when $Re_p < 1$, parameter defined as:

$$Re_p = \frac{d_p|U - U_p|}{\nu} \tag{2.18}$$

ν being the kinematic viscosity and $U$and U_p being respectively the fluid velocity and particle velocity.
This parameter, the particle Reynolds number, is a non-dimensional measure of a particle's relative velocity with respect to the surrounding fluid. It has an important role in parameterizing the momentum exchange between the particle and fluid phases.
When $Re_p < 1$ the flow around the particle is considered to be in the Stokes regime. As the Re_p increases, the flow separates to form a recirculating eddy in the wake of the particle. With

time, the wake of an isolated particle will become time-dependent and begin to shed vortices. Re_p values greater than 500 will result in turbolent flow. However, in this particular study, Re_p will leave the Stokes regime without reaching values significantly greater than 1.

The particle Reynolds number is a key factor in determining the amount of turbulence generated or dissipated by the particles, thus modulating the effect of turbulence. One example is that below a critical particle Reynolds number, the presence of particles tends to dissipate fluid phase turbulence, whereas at higher particle Reynolds numbers, turbulence is enhanced in particle-laden flows due to the vortex shedding mechanism [25].

It is advisable not to underestimate this issue by implementing a revised formula for $Re_p > 1$. This is done by introducing a correction factor C:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu C} \tag{2.19}$$

where:

$$C = 1 + \frac{Re_p^{2/3}}{6} \tag{2.20}$$

This is valid for $Re_p < 1000$, thus is a pretty generic form that can be tweaked for specific cases. By introducing this expression, it is noticeable that the ”Stokesian” particle depends on the characteristics of the carrier phase where it moves. Instead the correction factor takes into account the effect of inertial forces on the time relaxation of a ”non-Stokesian” particle. Therefore, in the case of a ”non-Stokesian” particle motion, its inertia depends also on the Re_p as described.

Another fundamental parameter for particles in turbulent flows is the Stokes number (*Stk*), which is an adimensional term that characterizes the particle inertia with respect to some or other scales of the flow. To deal with particle motion in a flow of gas with a gradient of averaged velocity along the longitudinal direction, it is necessary to take into account the inertia of particles when analyzing the process of relaxation of the flow. For this reason *Stk* is introduced.

There are various definitions of this number, given that its definition is the ratio between the particle relaxation time scale and the most appropriate fluid timescale based on the type of flow. In this case the parameter of dynamic inertia of particles in large-scale fluctuation motion is needed:

$$Stk = \tau_p \frac{v_f}{L} = \frac{\tau_p}{\tau_l} \tag{2.21}$$

with τ_l the characteristic time of the carrier phase, dependent on the fluid velocity v_f and characteristic length of the flow L.
It should be mentioned that this equation is valid only for Stokesian particles. When $Re_p > 1$ the Stokes values obtained underestimates the impact of the fluid drag force on the particle. Anyway, given the fact that in this work the limit will not be surpassed by much, Equation (2.21) is considered valid in any case.

2.4.1 Turbophoresis

Without getting too deep into the derivation of particle parameters, which will be continued later, it's better to give a little insight into the key phenomenon of the whole thesis: migration of particles.

An intriguing aspect of particle-laden flows is the preferential migration of the particles to certain areas within the fluid flow, which is often characterized by the Stokes number (Stk) allocated to each particle. At low Stk values, particles function as tracers and are distributed uniformly. At high Stk, particles are heavy and are influenced less by the fluid and more by its inertia. At intermediate St values, the particles are impacted by both the fluid's motion and its inertia, leading to several interesting behaviours. This is particularly noticeable in wall-bounded flows where there is a velocity gradient in close proximity to the wall.

One of the earliest studies discussing the phenomenon of preferential migration is the experiment conducted by Segre and Silberberg. Their findings indicate that a neutrally buoyant particle, placed in a laminar pipe flow, eventually reaches an equilibrium position between the wall and axis. This occurrence is commonly known as the Segré-Silberberg effect. Saffman provided an explanation for this phenomenon in terms of the force exerted on the particle when it encounters a velocity gradient.

Even non-neutrally buoyant particles exhibit comparable preferential migration. At low Stk, the particles tend to settle at an equilibrium position, while at high Stk, they begin to oscillate about the centre of the channel.

The phenomenon is particularly interesting in turbulent flows. Here the turbophoretic force causes a high concentration of particles near the walls. Studies conducted through both experimental and particle-resolved DNS means offer insight into the mechanism behind this process, in terms of the Saffman lift and the turbophoretic force.

Small, heavy particles in an in-homogeneous turbulent flow tend to migrate from regions of high turbulence intensity towards lower intensity regions. This is what its called turbophoresis, and it is driven by a differential in turbulent dispersion rates between the different regions of the flow.

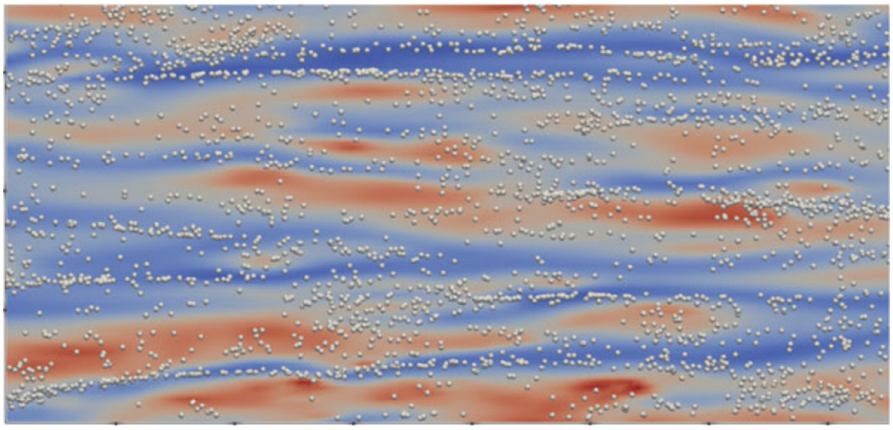


Figure 2.8: Example of turbophoresis; the blue region has low turbulence, the red one high turbulence[12].

Turbophoresis is the force exerted on particles when there is a gradient in turbulent kinetic energy (TKE).

There are two types of turbophoresis: global and local. When the diameter of the particle, is greater than the Kolmogorov scale of length η , such that it encounters a gradient in TKE, the particle ”feels” a resultant force in the direction of the lower TKE.

Turbulent fluctuations transfer momentum to the particle on one side faster than on the other side, resulting in a net force on the particle in the direction of decreasing TKE. The force's magnitude depends on the gradient of the TKE across the particle and on the size of the particle.This just described is the local turbophoresis.

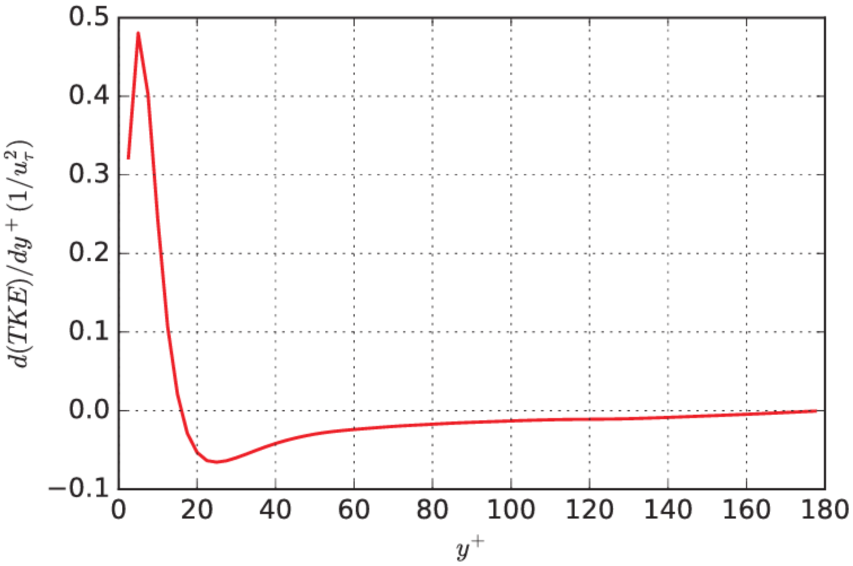


Figure 2.9: Gradient of turbulent kinetic energy in the channel [11].

Another way in which turbophoresis can act, even when particles are not resolved, is through ”global” turbophoresis. Turbulent flows will randomly disperse particles through turbulent eddies. If a particle possesses no inertia, signifying Stk = 0 (tracer particles), turbulence will guarantee homogenous particle concentration. Nevertheless, if particles have finite inertia, flow eddies must overcome the particles' inertia to move them. Areas with higher TKE would disperse particles more effectively, whereas areas with lower TKE would experience slower particle dispersion.

This can result in the concentration of particles in regions with low turbulence, causing them to aggregate in the proximity of the wall or the center. The random movement of turbulence will continuously transfer particles to either the wall area or the centre, where they will accumulate.

The global turbophoresis is dominant only on particles with moderate Stokes, since they respond well to turbulent fluctuations , while near 0 or at very high values it vanishes. Also, if the particles are bigger than the length scale, only the local one is relevant [11].

Back to the thesis case, when in wall-bounded turbulent flows, no-slip and no-penetration conditions cause turbulence intensities to vanish at solid boundaries resulting in sharp gradients of turbulence intensity in the viscous sublayer and buffer region. As a consequence, particles tend to accumulate in the viscous sublayer, at high concentrations relative to the surrounding flow. This can influence many of the physical process happening in the fluid, such as deposition, collision, thermal or radiation transmission and absorption.
This itself explains the importance of understanding this kind of phenomenon.

3

Methodology

This chapter discusses the methodology employed to study a turbulent particle-laden flow. Specifically, it focuses on an incompressible, wall-bounded flow analysed via RANS or LES techniques, which is later enriched with a range of rigid spherical particles, analysed via hybrid Langrangian-Eulerian approach. In Sections 3.1 to 3.3 it will be provided a short overview of the Navier-Stokes equations, simulation techniques and Wall Models. Following this, Sections 3.4 and 3.5 will provide a swift explanation of particle diffusion models and the Langevin model.

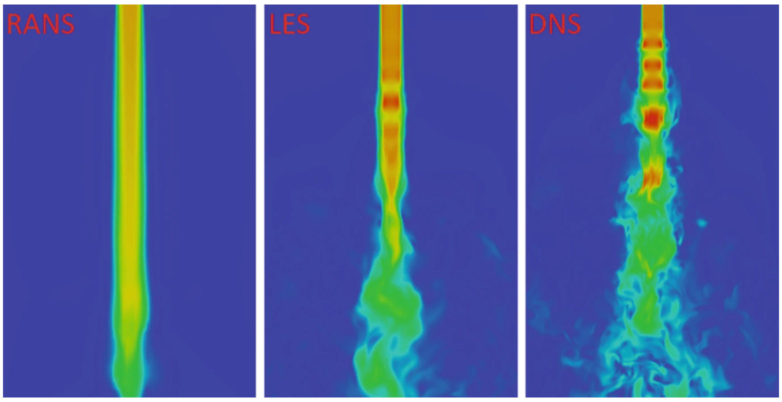


Figure 3.1: Velocity distribution for a turbulent jet using RANS, LES, and DNS[23].

3.1 Navier-Stokes Equations

To analyze and describe the behaviour of a flow, the Navier-Stokes equations are needed. In the case of an incompressible (ρ =cost) turbulent flow, the resultant continuity and momentum equations are [14, 2]:

$$\nabla \cdot \mathbf{u} = 0 \tag{3.1}$$

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \nabla \cdot (2\nu \mathbf{E}) \tag{3.2}$$

Here **u** is velocity vector of the fluid, ν is the kinematic viscosity, p is the pressure and **E** is a 3x3 tensor called the ”strain-rate tensor”. **E** is defined as:

$$\mathbf{E} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \tag{3.3}$$

The set of equations just defined is valid in any case, both turbulent and non-turbulent flow. Since turbulence comprises a fluctuating component, as mentioned earlier, implementing the Reynolds decomposition again is advisable. This will ensure that the symmetry properties remain observable, particularly in the context of mean fields. What is obtained is this system:

$$\nabla \cdot \langle \mathbf{u} \rangle = 0 \tag{3.4}$$

$$\frac{\mathrm{D}\langle \mathbf{u} \rangle}{\mathrm{D}t} = -\frac{1}{\rho}\nabla p + \nabla \cdot (2\nu \langle \mathbf{E} \rangle) - \nabla \cdot \tau_R \tag{3.5}$$

$$\tau_R = \langle \mathbf{u'} \cdot \mathbf{u'}^T \rangle \tag{3.6}$$

The equations are quite alike, with some alterations, such as the inclusion of fluctuating terms (indicated by the superscripts), and the presence of the mean fields, within the $\langle\rangle$ symbols, instead of the velocity vectors.

But there actually is an additional term, the Reynolds stress tensor τ_R: this one depends only from the fluctuation components in respect to the mean field, so that in a non-turbulent flow, when **u'** is zero, also τ_R will be, thus Equation (3.5) will be equal to Equation (3.2)

This additional stress tensor does not actually represent stresses, but instead symbolize the additional effect of the turbulence on the flux, including factors such as diffusivity and fluid mixing. In fact its value is never actually equal to zero and it is also usually greater than the viscous counterpart, except for in the wall region.

At this stage, the system is still not solvable: in fact the Reynolds stress tensor introduces a closure problem due to the auto-correlation of the **u'** terms. The only way to solve and use the RANS (the set of Equations (3.4) to (3.6)) is by introducing a mathematical model, of which there are various types. They will now be explained briefly.

3.2 Resolutive Methods

As already described, a mathematical barrier exists that renders this problem not resolvable using any analytical process, thus numerical approaches are necessary. Multiple methods are available for implementation, all of which fall under three categories. The choice of category primarily depends on computational cost and required accuracy.

It is evident that the more accurate process is needed, the more time it will take to achieve it. The same goes for the complexity of the geometry and the type of simulation. Frequently, there is no alternative but to strike a balance between time and these considerations.

3.2.1 DNS

The most straightforward resolutive method consists in discretizing the integration domain and subsequently solving the Navier-Stokes equations (Equations (3.1) and (3.2)) for each segment. This technique is known as Direct Numerical Simulation (DNS), because it enables the simulation of all processes in the turbulence by directly solving the Navier-Stokes equations, capturing all motion scales from the domain size to the Kolmogorov scale η. To be able to describe every phenomenon, the cellsize must be equal to that particular scale. Consequently, DNS often becomes infeasible due to its computational costs.

To give a scale of this problem, let's suppose that the domain scale lenght is L, while the grid spacing is Δx. The number of points in the x direction will be $N_x = L/\Delta x$.
From Kolmogorov's theory, the Kolmogorov scale and the turbulent kinetic energy are defined as:

$$\epsilon \propto \frac{U^3}{L} \qquad \eta \propto \frac{\nu^3}{\epsilon^{\frac{1}{4}}}$$

Considering the scale needed for Δx, the number of points needed in the three axis is:

$$N_p \propto N_x^3 \propto \left(\frac{L}{\eta}\right)^3 = \left(\frac{L}{\nu^3/\epsilon^{1/4}}\right)^3 \propto \left(\left(\frac{L^4 U^3}{L\nu^3}\right)^{3/4}\right)^3 \sim Re^{9/4}$$

Meanwhile, the frequency at which the simulation is needed to obtain a median field is given from:

$$N_{\Delta t} = \frac{T}{\Delta t} \sim \frac{T}{\tau_\eta} = \frac{L/U}{\tau_\eta} \propto Re^{1/2}$$

where $\tau_\eta = \frac{\eta}{u_\eta} = (\nu/\epsilon)^{1/2}$ is the Kolmogorv characteristic time.
Concluding, the total simulation time will be:

$$T_{tot} \propto N_p \cdot N_{\Delta_t} \propto Re^{11/4} \simeq Re^3$$

As evident as it is, the computional time is directly dipendent from the Reynolds number,

therefore DNS can be employed only in low *Re* cases, to avoid generating impractical times and costs. It's truly a pity, cause DNS would perfect to simulate large-scale flows with a high *Re*, where turbulence is dominant, with high reliability in the results.
In fact, DNS results are comparable to actual experimental ones.

3.2.2 RANS

A second possibility is to focus solely on the mean-field of the flow. This occurs with the RANS methods, whereby velocity fluctuations are not taken in consideration, if not for the introduction of models that describe them.
This means that the boundary conditions symmetries are restored, therefore in stationary conditions a 2D simplification is allowed. On the other hand, this kind of simulation can no longer be considered an experiment, and needs to be validated with reference data.

The major advantage derived from this technique is the reduced resolution needed to resolve the phenomena in the simulated flux, due to the fact that the quantity it is being referred to now is **U** and not **u**. Therefore RANS simulations are several magnitudes lighter in terms of computational time.
Still, to solve RANS equations the Reynolds stress tensor is to be determined, and this introduces more variables in the system: there are 4 equations in 10 unknowns, due to the tensor being symmetrical $(u, v, w, p, \tau_{xx}, \tau_{yy}, \tau_{zz}, \tau_{xy}, \tau_{xz}, \tau_{yz})$.
This leads to the already cited closure problem, because the tensor can not be determined analytically, but also not be ignored.
To get out of this impasse, the Boussinesq hypothesis on Turbulent viscosity (ν_t) is introduced:

$$\mathbf{T} = -p\mathbf{I} + 2\mu\mathbf{E} \tag{3.7}$$

$$\tau_R = -\frac{2}{3}\mathbf{KI} + 2\nu_t\langle\mathbf{E}\rangle \tag{3.8}$$

Starting from the viscous stress tensor T, τ_R can be modeled in a similar way, replacing the molecular viscosity μ with Turbulent viscosity ν_t. Clearly this two properties are different from each other, the first one being a property of the fluid, the other one a motion property. To complete the picture K and E are the turbulent and kinetic tensor.

At this point, the RANS equations are simplified to having just 5 unknowns; it is then introduced a model for ν_t.

For this purpouse there are many options: the most common ones are $k - w$,$k - \epsilon$ or else some of the blends between the main types, like $k - w - BSL$ or $k - w - SST$, as they are either efficient in the bulk or at the wall.

3.2.3 LES

The last technique, LES, tries to find an halfway between DNS and RANS, by solving only a certain part of the fluid.

While the larger scales are completely simulated like in DNS, as they are anisotropic and depend on geometry, the behaviour of the smaller scales are modeled, as they are actually universal. Also, as in the DNS case, the problem is still 3D and non stationary. This way LES represents a valid option in terms of computational cost, accuracy, and surely is most recommended in high Reynolds number cases.

To start, a threshold scale Δ needs to established, over which turbulence and non stationary effects will be simulated. Usually the procedure employed in LES techniques consists of three steps:

1. **Defining the filtering operation**, which means to determine the scale where the velocity field will separate in large or small scales. It is then defined the LES filtering operator (G_Δ), of which it's only needed to know that it allows the decomposition of velocity $u_i = \tilde{u}_i + u_i'$. The first component is the filtered velocity, which is related to the motion of larger scales, while the other is the residual, or subgrid, one.

2. **Filtered Navier-Stokes equations**. Applying the filter gives us this system:

$$\frac{\partial \tilde{u}}{\partial t} + \nabla \cdot \tilde{u}\tilde{u} = -\frac{1}{\rho} + \nabla \tilde{p} + \nabla \cdot (2\mu\tilde{E}) - \nabla \cdot \tilde{\tau}_R \tag{3.9}$$

$$\tilde{\tau}_R = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j \tag{3.10}$$

The last equation defines the Residual Stress Tensor, and similar to the RANS case, it is linked to turbulent structures effects, in particular it represents the effect of the unresolved structures of the flow on the filtered ones. The found system is actually very similar to the RANS one (except for first one actually representing 3D flow). Unfornately this means that as for the Reynolds stress tensor (τ_R), also the Residual stress tensor ($\tilde{\tau_R}$) needs a closure model.

3. **LES Closure Model**. As in the RANS case, there are still 10 unknowns in only 4 equations. Firstly, the tensor trace is introduced, which equals $2\tilde{k}_r$ for $\tilde{\tau_R}$, and is substituted inside the equations, leaving only 9 unknowns: $\tilde{u}_i$,$\tilde{p}$ and five independent τ_r components.
   Now a model to describe the deviatoric part of the tensor needs to be introduced, the Smagorinksi Model: it introduces the eddy viscosity concept, and consequently the Residual viscosity (ν_r); this approach treats dissipation of kinetic energy at sub-grid scales as analogous to molecular diffusion, where the deviatoric part of τ_r is now expressed as:

   $$-\tau_r = 2\nu_r \tilde{S}_{ij} \tag{3.11}$$

   where $\tilde{S}_{ij}$ is the residual strain rate tensor. Now only an expression for ν_r is needed:

   $$\nu_r = (C_S \Delta)^2 |S| \tag{3.12}$$

   Therefore, in the Smagorinski Model the eddy viscosity is proportional to the subgrid characteristic length scale Δ, and to the module of the local strain-of-rate. The Smagorinski constant C_S is determined so as to dissipate the appropriate level of energy at the grid spacing utilized: the value found is 0.12 ± 0.06.

After these passages the filtered equations can be solved for the filtered velocity.
The results obtained using this classical Smagorinski approach are not actually very true to the actual phenomena happening in the flux.

To be more precise, it becomes problematic in the flow near walls and solid boundaries, where quasi-streamwise vortices become more and more dominant: these near-wall eddies span from approximately $l \sim O(10^2)$ and grow with the increase of the distance to the wall; as Friction Reynolds number (Re_T) increases, these eddies will decrease in size relative to the boundary-layer thickness, risking a conflict with the resolution requirements[6].

This problem, and some other inconsistencies between ν_r never being null and the fact that turbulence should be completely damped at the wall, makes it necessary to find a better solution for a study focused on the near-wall region.

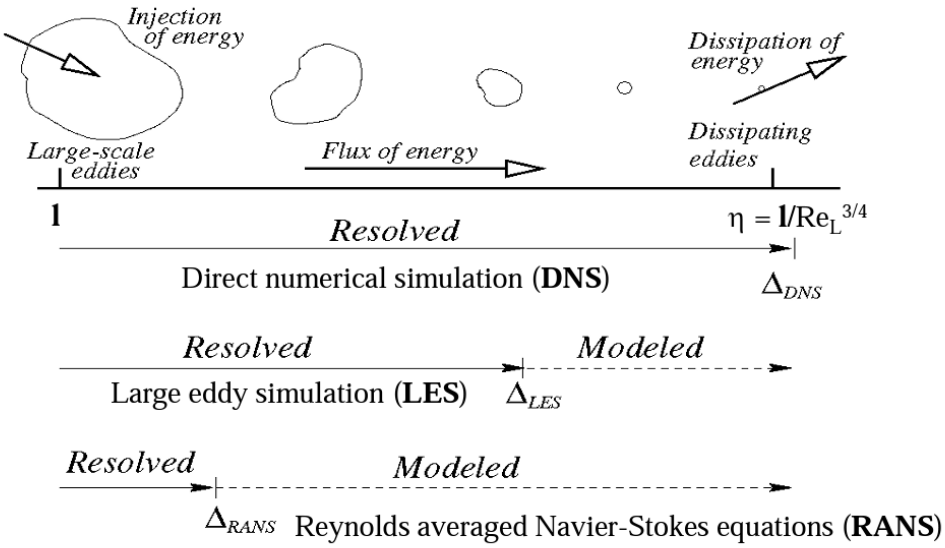


Figure 3.2: Grid resolution for the main cited methods [1].

3.3 Wall-Modeled LES

To improve the results of LES in the boundary layer, there are two possible methods:

- Wall-Resolved LES (WRLES): where near-wall eddies are resolved on the computational grid
- Wall-Modeled LES (WMLES): where near-wall eddies are modeled on the computational grid

Between the two options, WRLES is significantly similar to a DNS approach. Therefore a Wall Model is preferred: in this kind of approach, to properly replicate the near-wall eddies effect, an eddy viscosity ν_τ will be added close to the wall.

One of the possible formulations, the Wall-Adapting Local-Eddy viscosity (WALE), is:

$$\nu_\tau = (C_w\Delta)^2 \frac{\left(S^d_{ij}S^d_{ij}\right)^{3/2}}{\left(\bar{S}_{ij}\bar{S}_{ij}\right)^{5/2} + \left(S^d_{ij}S^d_{ij}\right)^{5/4}} \tag{3.13}$$

As explained by Nicoud [17], here Δ is still the grid spacing, C_w is a different constant, obtained assuming that the WALE model gives the same ensemble-average subgrid kinetic energy dissipation as the classical Smagorinsky model, and $\bar{S}_{ij}$ is the symmetric velocity part of the velocity gradient.

The values of ν_t computed this way still have to let **u** mantain the correct velocity profile, which in the case of this work will be:

$$\frac{u}{u_\tau} = \frac{1}{k} log(1 + ky^+) + C_k \left(1 - e^{-y^+/11} - \frac{y^+}{11} e^{-0.33y^+}\right) \tag{3.14}$$

This profile interpolates the log-law along the whole boundary layer. The two new constants are k=0.41 and C_k=7.8.

At this point the obtained eddy viscosity can be added to the physical viscosity when required: $\nu_{tot} = \nu + \nu_t$. Thus obtaining a more realistic model regarding the wall boundary layer.

3.4 Particle Diffusion

After examining the fluid aspect of the system, it is important to introduce the methods by which the behaviour of the particles will be addressed in the course of this work.

When studying the motion of a flow, two kind of approaches can be taken: Eulerian approach, where the flow properties are probed at fixed locations of the spatial domain, and the Lagrangian approach, that on the contrary deals with every single particle while calculat-

ing their trajectories individually.
While it is simpler to measure flow properties at fixed locations, it is more intuitive to describe a flow by examining at the trajectories of fluid tracers. So, in the case of this work, when comparing fluid tracers to the particles studied, it is appropriate to discuss the Lagrangian approach. This choice is also supported from the fact that the Lagrangian approach becomes heavy from a computational point of view only when treating a particularly large number of elements.
In this situation, it is logical to employ a Lagrangian approach, given the already imposed condition on the number of particles due to the necessity of using a one-way approach simulation.

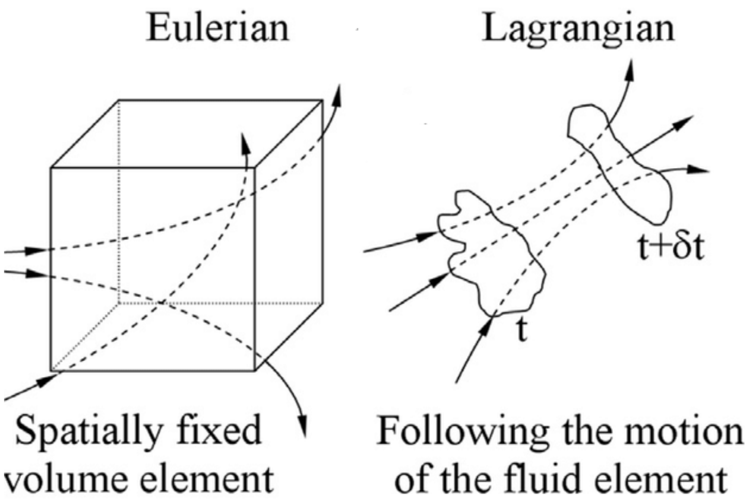

Eulerian
Lagrangian
t+δt
t
Spatially fixed volume element
Following the motion of the fluid element

Figure 3.3: Eulerian and Lagrangian representation of fluid flow equations[24].

Having established this, it is possible to move on to the treatment of particles.
The particles suspended in the turbulent flow are considered hard rigid spherical bodies, which behave elastically when hitting other elements of the environment.
The particles can also be considered diluted as that particle-particle interactions are not significant. The dispersed phase can be considered as point-wise spherical particles which are entrained in a turbulent flow at isothermal conditions.

As summarized by Dehbi [7], if no simplifications are made, a series of different forces will be acting on every single particle: other than the flow itself, there will be the drag force and the lift force action, due to the spherical shape of the particle, and also the gravity can be taken into consideration. Usually the lift force is neglected, because most of the times the particle is much heavier than the carrier phase.

The vector force balance obtained is written as follows:

$$\frac{dU_p}{dt} = F_D(U - U_p) + g\left(1 - \frac{\rho_f}{\rho_p}\right) \tag{3.15}$$

with F_D being the drag force per unit mass, also expressed as:

$$F_D = \frac{18\mu}{\rho_p d_p^2} C_D \frac{Re_p}{24} \tag{3.16}$$

In the above equations, U is the fluid velocity, U_p the particle velocity, ρ_p the particle density, ρ_f the fluid density, d_p the particle diameter, g the gravity acceleration, μ the fluid molecular viscosity, and Re_p the particle Reynolds number, defined as:

$$Re_p = \frac{d_p|U - U_p|}{\nu} \tag{3.17}$$

The drag coefficient can be computed via analytical equations, by introducing a series of constants that apply to spherical particles for wide ranges of Re_p. An example of these equations is:

$$C_D = \beta_1 + \frac{\beta_2}{Re_p} + \frac{\beta_3}{Re_p^2} \tag{3.18}$$

To conclude, the trajectory $x(x_1, x_2, x_3, t)$ of the particle is obtained by integration of the following velocity vector equation with respect to time:

$$U_p = \frac{\mathrm{d}x}{\mathrm{d}t} \tag{3.19}$$

This expressions alone are enough to compute the trajectory of individual particles in laminar flows. The particle concentration and deposition rates are deduced in a deterministic way. Particle dispersion is accurately predicted.

When turbulence is considered in the equation, the computation of particle dispersion becomes significantly more involved as the random velocity fluctuations do not permit a deterministic knowledge of particle trajectories. It is then necessary to resort to stochastic computations to address the ”average” particle dispersion. There is not only a single way to resolve this issue.
There are therefore models that attempt to simulate turbulence using complementary equations, and calculate the instantaneous turbulent velocities starting from local quantities, like the mean turbulent kinetic energy, the time scale and the distance to the wall.
These techniques, called random walk models, are relatively easy to implement and also reasonable computationally wise. One of them is the Discrete Random Walk (DRW), where the turbulent dispersion of particles is modeled as a succession of interactions between a particle and eddies which have specific lengths and lifetimes. The particle is captured from an eddy that has velocity produced by the sum of the mean flow and a random ”instantaneous” velocity, which is at times constant in time. When the lifetime of the eddy is over, the particle ”jumps” to a new eddy, and so on.
A more physically accurate depiction of fluid turbulence is given by Continuos Random Walk (CRW) models, as they represent the instantaneous velocities in a continuous way. CRW models, which are usually based on the Langevin equation, show more realistic predictions of turbulent particle dispersion in flows where inhomogeneous effects are important, like in this thesis case.
Before explaining how the model is implemented, it is necessary to explain the concept at its basis, the one just cited above: the Langevin equation.

3.5 Langevin Model

All the information reported in this section is taken from Rodean [22].

In physics, a Langevin equation is a stochastic differential equation which describes a system evolution when subjected to a combination of deterministic and fluctuating forces. The former ones have a behaviour of changing slowly in comparison to the latter ones. The microscopic, fast variables are the ones responsible for the stochastic nature of the Langevin equation. One of its applications is to Brownian motion, particularly relevant to the case study, given it models the fluctuating motion of a small particle in a fluid.

The use of this equation as a model for turbulent diffusion is done on the basis that the Fokker-Planck equation is the Eulerian equivalent of the Langevin equation, which is Lagrangian.

The most basic form of the Langevin model resulting from it, is this:

$$du_i = a_i(\mathbf{x}, \mathbf{u}, t)dt + b_{ij}(\mathbf{x}, \mathbf{u}, t)dW_j(t) \tag{3.20}$$

$$d\mathbf{x} = \mathbf{u}dt \tag{3.21}$$

Here **u** is the Lagrangian velocity of any marked particle, *a* is the deterministic term, the drift, *b* is a stochastic term, the diffusion and dW is a Gaussian distribution.

The basic form of the Langevin equation can be derived relatively easily through physical reasoning. Specifically, the particle's time-dependent velocity in space is a function of the drift and diffusion, which correspond to the first and second moments of Equation (3.20), combined with the characteristics of the incremental process that is modelled using a Gaussian distribution.

Also, each timestep only depends on the previous one, making this model a Markov chain. Therefore, this stochastic model describes a sequence of possible events where the probability of any event solely relies on its preceding event. As a result, the particle motion's ”story” is eliminated from any given instant's influence.

The process to obtain the two components of the equation is pretty verbose, so their derivation will be kept as simple as possible.

Starting with the stochastic term, b is kept independent from the velocity u, to keep the model simple.

To obtain the expression that describes it, the first step is the introduction of the Lagrangian structure function, which is the ensemble average of the square of the change in the Lagrangian velocity in the time interval Δt:

$$D(\Delta t) = \langle |v(t + \Delta t) - v(t)|^2 \rangle = \langle (\Delta v)^2 \rangle \tag{3.22}$$

For Kolmogorv's inertial subrange $\tau_\eta \ll \Delta t \ll \tau$ where τ_η is the Kolmogorov time scale and τ is the fluid time scale:

$$D(\Delta t) = C_0 \epsilon \Delta t \tag{3.23}$$

where C_0 is a universal constant, with value between 2 and 5 for turbulence in the boundary layer, while ϵ is the ensemble-average rate of dissipation of turbulent kinetic energy. The coefficient b, or the stochastic component, can be related to C_0 by taking the ensamble average of Equation (3.20). Assuming a and b constant during a single timestep Δt applying a Wiener process the result is:

$$\langle (\Delta v)^2 \rangle = b^2 \Delta t = |C_0 \epsilon(y)|^{1/2} \tag{3.24}$$

This equivalence is then used in the first form of the Langevin equation, giving:

$$dw = a(y, v)dt + |C_0 \epsilon(y)|^{1/2} dW(t) \tag{3.25}$$

To satisfy the well-mixed condition, which means that the if the particles are initially well-mixed, they will stay that way, a certain condition on the Fokker-Planck equation equivalent of Equation (3.25) has to be respected. This condition concerns the probability density function P_a in the y-v space of the particles of the fluid. This leads to the condition:

$$a(y, v) = \frac{C_0 \epsilon(z)}{P_a(y, v)} \frac{\partial P_a(y, v)}{\partial v} + \frac{\Phi(y, v)}{P_a(y, v)} \tag{3.26}$$

where the distribution Φ satisfies:

$$\frac{\partial \Phi(y,v)}{\partial v} = -\frac{\partial}{\partial y}|vP_a(y,v)| \tag{3.27}$$

The process to obtain this distribution is here skipped as it exhudes the thesis scope.

What is obtained is:

$$dv = \left(-\frac{C_0\epsilon Q}{P_a} + \frac{\Phi}{P_a}\right)dt + (C_0\epsilon)^{1/2}dW(t) \tag{3.28}$$

where Φ and Q are expressed via terms enclosed in the process skipped just before.
In Equation (3.28) the first component of the a term is a "fading memory", while the second component is a drift term to account for the vertical inhomogeneity of the turbulence. This is much clearer in the case of a Gaussian inhomogeneous turbulence:

$$\frac{C_0\epsilon Q}{P_a} = \frac{C_0\epsilon}{2\sigma_v^2}v \tag{3.29}$$

$$\frac{\Phi}{P_a} = \frac{1}{2}\left(1+\frac{v^2}{\sigma_v^2}\right)\frac{\partial \sigma_v^2}{\partial y} \tag{3.30}$$

giving:

$$dw = \left[-\frac{C_0\epsilon}{2\sigma_v^2}v + \frac{1}{2}\left(1+\frac{v^2}{\sigma_v^2}\right)\frac{\partial \sigma_v^2}{\partial y}\right]dt + (C_0\epsilon)^{1/2}\,dW(t) \tag{3.31}$$

The fading memory term in Equation (3.29) comes very naturally from the use of the use of the Fokker-Planck equation and the Lagrangian structure function used to solve the a and b terms at the beginning.

When in stationary, homogeneous Gaussian turbulence conditions, the above mentioned fluid timescale τ can be written as:

$$\tau = 2\sigma_v^2/C_0\epsilon \tag{3.32}$$

This can be substituted inside Equation (3.29) to give a simpler form for the "fading" term, and from that Equation (3.31) can be written as:

$$dv = -\frac{v}{\tau_L}dt + \left(\frac{2\sigma_v^2}{\tau_L}\right)^{1/2} dW(t) \tag{3.33}$$

where **v** is now the vertical velocity, σ us the variance of the updraft or downdraft velocities and τ_L is the fluid time scale.
This form is valid for high Reynolds flows with three-dimensional turbulence, and for timestep values higher than the Kolmogorov time scale τ_k. Also the random forcing $dW(t)$ is Gaussian.

It is possible to give a reinterpretation of the physical meaning of the Langevin equation as model for turbulent diffusion, which is relevant to the thesis case:

$$\frac{du}{dt} = -\alpha u + \beta\xi(t) \tag{3.34}$$

As already seen, **u** will be the Lagrangian velocity fluctuation from the mean motion of the particle, the α coefficient is not related to fluid viscosity for Brownian motion, but it is rather a measure of the above mentioned "fading memory" of the particle velocity. As for, the product of the coefficient β and the random function $\xi(t)$ does not represent irregular acceleration from molecular bombardment as in Brownian motion, instead it represents the random acceleration from pressure forces with short correlation times on the order of the Kolmogorv time scale τ_k.

The equation just described is still a differential equation, and can be solved as it was already done with the generic case. By introducing the Wiener process:

$$W(t) = \int_0^t \xi(s)ds \tag{3.35}$$

and some other steps similar to the ones already done, Equation (3.33) is obtained. This form is obtained with the assumption that the turbulence is both stationary and homogeneous. This does not apply to all cases: it is not unusual that the velocity variance profile is non-uninform. A drift correction term is required. Its presence gives a new version of the equation:

$$du = \left(-\frac{u}{\tau_L} + 2\sigma_u \frac{2\sigma_u}{dz}\right) dt + \sigma_u \left(\frac{2}{\tau_L}\right)^{1/2} d\xi(t) \tag{3.36}$$

A better explanation on this derivation and its implications will be given along the model implementation, in Chapter 4.

4

Implementation

It is now time to implement the model chosen to reproduce the particle behaviour. Like many models, this one also necessitates an ad-hoc formulation for the specific circumstances.
The starting points for its development were three papers: Dehbi [8], Marchioli et al. [16] and Dehbi [7]. These three papers deal with practically the same problem, as they all start from exactly the same DNS data of a turbulent flow, and then implement some sort of Langevin equation approach to integrate the particle behaviour in the system.
It all stems from Marchioli et al. work, which was the result of an international collaborative test case related to the production of a direct numerical simulation and Lagrangian particle tracking database for turbulent particle dispersion at low Re.

The idea was to provide a homogeneous source of data relevant to the general problem of particle dispersion in wall-bounded turbulent flows. In that work, many different numerical approaches and computational codes have been used to simulate the flow.
Others have since extended the study, first Dehbi [7] with the derivation and integration of the Langevin model, and again Dehbi [8] with the implementation of a hybrid Lagrangian-

Eulerian approach and a simplification of the model.
All of this is meant to validate and integrate the dataset of DNS simulation and the simplified models.

In this context, the present work aims at replicating the original results with less powerful means to determine if the model can still be considered valid in simpler applications and, hypothetically, if the model can withstand other complications even with simpler initial assumptions.
Section 4.1 will outline the features of the system to be simulated. Section 4.2 will detail how to derive some of the required Eulerian parameters, while Section 4.3 will present the executed version of Langevin's model. Section 4.4 will showcase all the essential code adaptations. Finally, Section 4.5 will present the obtained outcomes, compared to both the original DNS and Debhi's outcomes.

4.1 System Definition

The objective of this chapter is to reproduce the findings from the literature accurately. Therefore, it is appropriate to reproduce the simulated system correctly.
Particles are dispersed within a fully developed turbulent flow of air. The fluid flows between two parallel walls that extend infinitely. The governing equations for the fluid, in dimensionless form, are:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{4.1}$$

$$\frac{\partial u_i}{\partial t} = -u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{Re_\tau} \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial p}{\partial x_i} + \delta_{1,i} \tag{4.2}$$

Here u_i are the components of the dimensionless velocity vector, $\delta_{1,i}$ is the mean dimensionless pressure gradient that drives the flow, p is the fluctuating kinematic pressure, and $Re_\tau = u_\tau h/\nu$ is the shear Reynolds based on the friction velocity and on the half channel height, that from now on will be called h. The shear velocity derives from $u_\tau = \sqrt{\tau_w/\rho}$ with τ_w being the wall shear stress.

The periodic channel simulating the infinite parallel wall channel has a half-height h of $0.02m$, a width of $2h$ and a length of $4h$. Anyway, the simulation will consider dimensions as dimensionless quantities, allowing for the obtained solutions to be rescaled according to the specific case. The density of air is taken to be $1.3\ kg/m^3$ and it's kinematic viscosity $15.7 \times 10^{-6} m^2 s^{-1}$. There will be periodic boundary conditions enforced in the axial and spanwise directions. The geometry of the channel consists of two infinite flat parallel surfaces, the origin of the channel is found at the center of the system: the x-,y- and z- axes correspond to the streamwise, wall-normal and spanwise directions. No-slip boundary conditions are applied at the walls.
This characteristics of cyclic motion should allow the manifestation of all the phenomena that are expected to show.

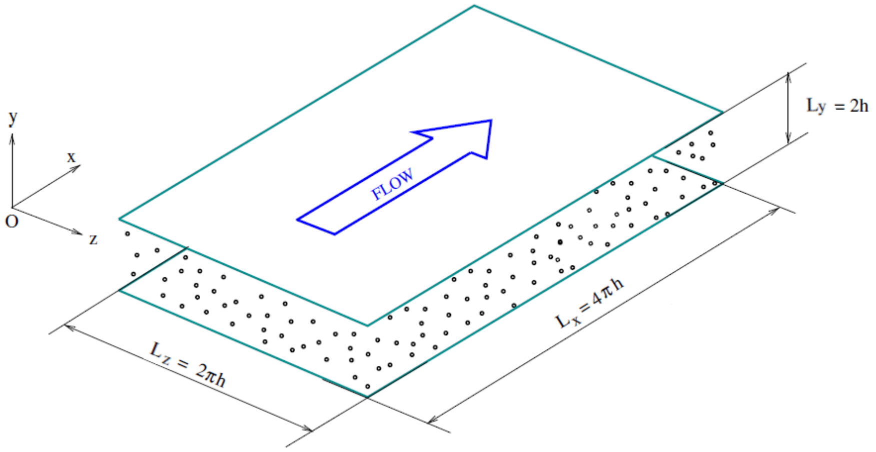


Figure 4.1: Particle-laden turbulent gas-flow in a flat channel: computational domain[16].

The mean axial fluid velocity is $U_b = 1.65m/s$ with a corresponding friction velocity $u_\tau = 0.11775m/s$. The Reynolds number based on the bulk velocity is thus $Re_b = 2280$, whereas $Re_\tau = 150$. The channel height will correspond to 300 wall units.

The particles injected in the flow have density of $\rho_p = 1000kg/m^3$ (suspended water particles), and are at a density low enough to consider dilute system conditions, as described in

Section 3.4.

Therefore one-way coupling approach is sufficient to describe the problem, and particle-particle interactions are insignificant. As stated, particles are assumed to be rigid, spherical and pointwise.

The particle motion is described by ordinary differential equations for particle velocity and position at each time step. It is shown that for particles much heavier than the fluid, the only significant force is the Stokes drag.

The particle velocity Equation (3.15) is thus reduced to:

$$\frac{dU_p}{dt} = F_D(U - U_p) \tag{4.3}$$

In fact, the Brownian diffusion, mentioned here only for completeness, is ignored due to the particles diameters being greater than μm, the lift force is neglected due to the high ratio between particle and fluid density, but also the buoyancy, or the gravity factor, will not be considered.

All the other elements featured in the equations derived from Equation (4.3) will remain the same as described in Section 3.4, which means:

$$F_D = \frac{18\mu}{\rho_p d_p^2} C_D \frac{Re_p}{24} \tag{4.4}$$

$$Re_p = \frac{d_p |U - U_p|}{\nu} \tag{4.5}$$

In this implementation the drag coefficient C_D will be given by:

$$C_D = \frac{24}{Re_p}\left(1 + 0.15 Re_p^{0.687}\right) \tag{4.6}$$

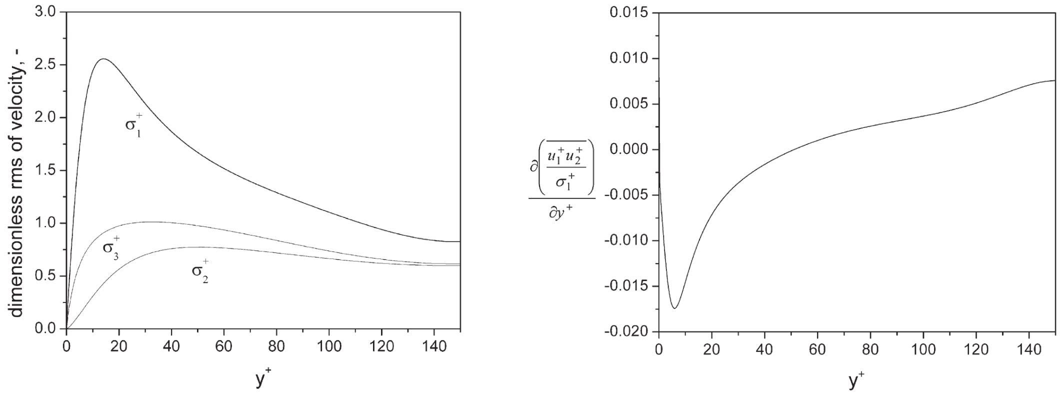
This coefficient contains the formulation saw in Section 2.4 for the correction C on the τ_p definition.

4.2 Specification of Eulerian rms and time scales

To integrate the Langevin equations, some Eulerian statistics are needed to complete the problem.

First of all, it is necessary to have the Eulerian standard deviation σ of velocity: these are obtained from fits of the DNS data obtained by Marchioli et al.. This fits are given by ratios of polynomials of order of 3 or 5, and agree to the original data with a correlation coefficient greater than 0.99 [8].

In the figures y^+ is the wall distance in dimensionless units, defined as $y^+ = \frac{yu_\tau}{\nu}$, where y is the particles distance to the nearest wall.



(a) Rms of fluid velocities[8]. **(b)** Rms of streamwis drift correction factor[8].

Figure 4.2: Fits derived from Marchioli DNS data.

The quantity τ_L, the Lagrangian fluid time scale, used in the velocity fluctuation equations, is defined in terms of the Lagrangian integral time scale T_L. If in the streamwise direction, T_L can be computed in terms of Lagrangian fluid autocorrelations:

$$R_1(\tau) = \frac{\overline{u_1(t) \cdot (t+\tau)}}{u_1(t) \cdot u_1(t)} \tag{4.7}$$

This term is the autocorrelation factor for the streamwise component, evaluated for every time step by ensemble averaging over a large number of particles. The step to get T_L is:

$$T_L = \int_0^T R_1(\tau) d\tau \tag{4.8}$$

This would be extremely complicated to use in the simulation, so a series of fits obtained from DNS calculations in the boundary layer showed that the Lagrangian integral time scales in all directions can be described by these analytical equations:

$$T_L^+ = 10 \qquad \mathit{for} \quad y^+ \leq 5 \tag{4.9}$$

$$T_L^+ = 7.122 + 0.5731 \cdot y^+ - 0.00129 \cdot {y^+}^2 \qquad \mathit{for} \quad 5 < y^+ < 200 \tag{4.10}$$

and from these derive the quantity:

$$T_L = T_L^+ \cdot \frac{\nu}{(u_\tau)^2} \tag{4.11}$$

Now all the quantities needed to compute the Lagrangian fluid time scale τ_L seen by a particle have been calculated. It is peculiar how the particle inertia needs to be taken into consideration to calculate a fluid time characteristics:

$$\tau_L = \frac{T_L}{\beta} \left(1 - (1 - \beta) \left(1 + \frac{\tau_p}{T_E} \right)^{-0.4\left(1+0.01\frac{\tau_p}{T_E}\right)} \right) \tag{4.12}$$

In this formula β is the ratio T_L/T_E, with the latter being the fluid Eulerian integral time scale. The ratio is taken to be 0.356. From this last equation, it can be noted that τ_L tends to T_L for particles with low values of the Stokes number, while for very inertial particles, it tends to T_E.

4.3 Langevin Implementation

The adaptation of the Langevin model is very well explained in Dehbi [7], thus most of the thesis version is based on Dehbi's work.
As previously said, the time history of the fluctuations in the carrier fluid that a particle sees as it moves in a flow, determines to a large extent its dispersion and deposition characteristics in turbulent fields. The most used method to describe fluid velocity fluctuations in homogeneous turbulence is the Langevin equation.
The change in particle velocity with time is assumed to be composed of a damping term, which is proportional to the velocity, and of a random forcing term with zero mean.

This concept is then extended to fluid velocity fields, transforming the Langevin equation into a stochastic differential equation which uses Markov chains to specify a possible increment du_i in the fluid velocity fluctuation:

$$du_i = -u_i(t)\frac{dt}{\tau_i} + \sigma_i\sqrt{\frac{2}{\tau_i}} \cdot d\xi_i \tag{4.13}$$

This will be added in the usual incremental displacement dx_i:

$$dx_i = (U_i + u_i)\, dt \tag{4.14}$$

Here τ_i is a timescale, σ_i the fluctuating standard deviation of velocity (=$\sqrt{\overline{u_i^2}}$), and $d\xi_i$ a succession of uncorrelated random numbers with zero mean and variance dt. Usually the distribution is a Gaussian.

What is known is that this model is not entirely accurate and requires correction to validate its utility. Specifically, while the Langevin equation can model homogeneous turbulence where all values are position-independent, it falls short in the context of a wall-bounded flow, where the turbulence is highly inhomogeneous and anisotropic in the boundary layer, necessitating modifications.

Numerous attempts confirmed that the wall-normal equation of the velocity fluctuations requires the inclusion of a certain δu_i mean drift correction to ensure that particles, on average, follow streamlines in the flow. Without this factor, DRW and CRW simulations exhibit errors up high as 550%.

The correction can be shown to be necessary with an analysis based on the decomposition of the instantaneous acceleration a_i:

$$a_i = U_j \frac{\partial U_i}{\partial x_j} \tag{4.15}$$

where the Einstein convention of summing up over repeated indices is used. Expressing the instantaneous velocity as the sum of the mean and fluctuating parts, remembering $\overline{u_i} = 0$:

$$U_i = \overline{U_i} + u_i \tag{4.16}$$

By unifying these last two equations, and averaging over time, after algebraic manipulation, the result is:

$$\overline{a_i} = \overline{a}_{i,mean} + \overline{a}_{i,drift} = \overline{U_j} \frac{\partial \overline{U_i}}{\partial x_j} + \overline{u_j \frac{\partial u_i}{\partial x_j}} \tag{4.17}$$

The mean acceleration of a fluid particle can therefore be broken down into a components: one is due to the mean flow, and the other is due to random turbulent fluctuations in an inhomogeneous flow field. The acceleration of drift produces a corresponding drift velocity which must be incorporated into the Langevin equation (Equation (4.13)) to account for turbulence inhomogeneities:

$$\delta u_i = \overline{u_j \frac{\partial u_i}{\partial x_j}} \cdot dt = \frac{\partial \overline{u_i u_j}}{\partial x_j} \cdot dt \tag{4.18}$$

The last equality requires the assumption of a divergency-free fluctuating velocity field, which is rational for this type of incompressible flows.

The kind of correction will depend on the specific flow conditions. Looking at the wall-normal direction of the boundary layer, the fully-developed flow assumption results in the wall normal derivate of $\sigma^2 = \overline{u_2 u_2}$ being the drift correction in the Langevin equation.

What has just been described needs a little more adjustment to apply to all particles, not just tracer particles, because as already mentioned they do not follow the fluid streamlines exactly. The easiest way to do that is described by Iliopoulos et al. [10]: by assuming that the fluid velocity seen by inertial particles is the same as the one seen by fluid particles, it is possible to say that the drift correction derived in Equation (4.18) apply to them at least to a first approximation.

However, a more rigorous method can be used, that applies to particles with arbitrary inertia: using the instantaneous acceleration of a fluid particle along the path of an inertial particle, the drift correction for the inertial particle can be obtained from the simpler drift correction of the fluid particle through a multiplicative factor:

$$\delta u_i = \frac{\partial \overline{u_i u_j}}{\partial x_j} \cdot \left(\frac{1}{1 + Stk} \right) dt \tag{4.19}$$

Where Stk is the parameter already defined beforehand, $Stk = \tau_p / \tau_l$.
Here τ_l will be obtained as described in Section 4.2, while τ_p is obtained as described in Section 2.4. In this particular case the formulation will be:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu} \qquad \text{if} \quad Re_p \leq 1 \tag{4.20}$$

and:

$$\tau_p = \frac{4}{3} \frac{\rho_p}{\rho_f} \frac{d_p^2}{C_D \left| U - U_p \right|} \qquad \text{if} \quad Re_p > 1 \tag{4.21}$$

Theoretically, it follows from this equation that the particle drift correction behaves correctly at the extremes, which means that for very large Stokes numbers the fluctuations and the particle motion result more decoupled, hence the drift correction tends to zero, while when Stokes tends to 0, the correction tends to the Equation (4.18) format.

4.3.1 Langevin in the boundary layers

In this section the Langevin model will be adapted to the boundary layer conditions.
The domain is assumed to be entirely comprised in the boundary layer. Turbulence is thus inhomogeneous in the wall normal direction, and hence the normalized Langevin equation is better suited than the classical formulation. The generic form will be:

$$d\left(\frac{u_i}{\sigma_i}\right) = -\left(\frac{u_i}{\sigma_i}\right) \cdot \frac{dt}{\tau_L} + d\eta_i + A_i dt \tag{4.22}$$

Here u_i is the fluid fluctuating velocity component, σ_i the rms of velocity, τ_L the Lagrangian time scale, $d\eta_i$ a succession of uncorrelated random forcing terms and A_i the mean drift correction that ensures the well-mixed condition (well-mixed particles will remain so as time goes by). In Equation (4.22) the increment represents the change in value of the fluctuations around the average, not the increment in the instantaneous velocity. To get the change in instantaneous velocity, it is required to include the term due to the mean velocity gradient, that is: $-u_j \delta\overline{U_i}/\delta x_j$, where $\overline{U_i}$ is the time averaged velocity in the ith direction. Since it is considered a boundary layer flow, the last term is non-zero only in the streamwise direction. The term $-u_2\left(\delta\overline{U_1}/\delta x_2\right)$ is thus added in that direction after u_1 is computed from Equation (4.22).
As shown in Dehbi [8] the obtained normalized Langevin equations are:

$$\begin{aligned} d\left(\frac{u_1}{\sigma_1}\right) &= -\left(\frac{u_1}{\sigma_1}\right) \cdot \frac{dt}{\tau_L} + \sqrt{\frac{2}{\tau_L}} \cdot d\xi_1 + \frac{\partial \frac{\overline{u_1 u_2}}{\sigma_1}}{\partial x_2} \cdot \frac{dt}{1+Stk} \\ d\left(\frac{u_2}{\sigma_2}\right) &= -\left(\frac{u_2}{\sigma_2}\right) \cdot \frac{dt}{\tau_L} + \sqrt{\frac{2}{\tau_L}} \cdot d\xi_2 + \frac{\partial \sigma_2}{\partial x_2} \cdot \frac{dt}{1+Stk} \\ d\left(\frac{u_3}{\sigma_3}\right) &= -\left(\frac{u_3}{\sigma_3}\right) \cdot \frac{dt}{\tau_L} + \sqrt{\frac{2}{\tau_L}} \cdot d\xi_3 \end{aligned} \tag{4.23}$$

where the $d\xi_i$s are modeled as a series of uncorrelated Gaussian random numbers with zero mean and variance dt.

4.4 Code Realization

Now that all the most important segments of the code have been described, it is possible to illustrate some of the particularities of the algorithm.

The code is implemented using the GFortran compiler, using Fortran 90 as language: Fortran 90 is a compiled imperative programming language, which is the basis of a large number of numerical simulation software used today. The algorithm will be structured like this:

Particle Definition and Positioning

In the first step a number of particles is generated in random positions of the fluid domain, making sure that no particles is positioned at its edges, to avoid any problem with the compiler. Every single particle property and characteristic has to be brought to zero before the start of the main block.

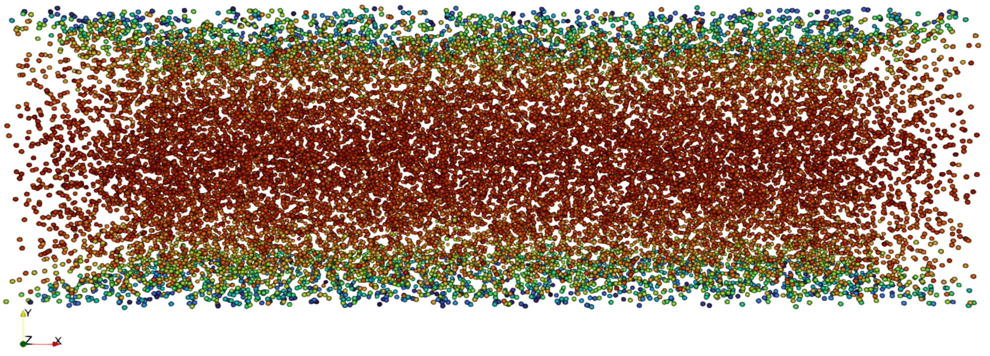


Figure 4.3: First iteration of the simulation: the particles are starting to gain velocity (red=fast, blue=slow, but not yet to migrate.

Main Block: Wall coordinate and Turbulent velocity

Every single particle has to be simulated individually for each time step. The first step for each of them will be the attribution of the corresponding wall coordinate, necessary in basically all the steps of the algorithm. Also important is the attribution of the local fluid velocity seen by the particle, obtained through Equation (3.14).

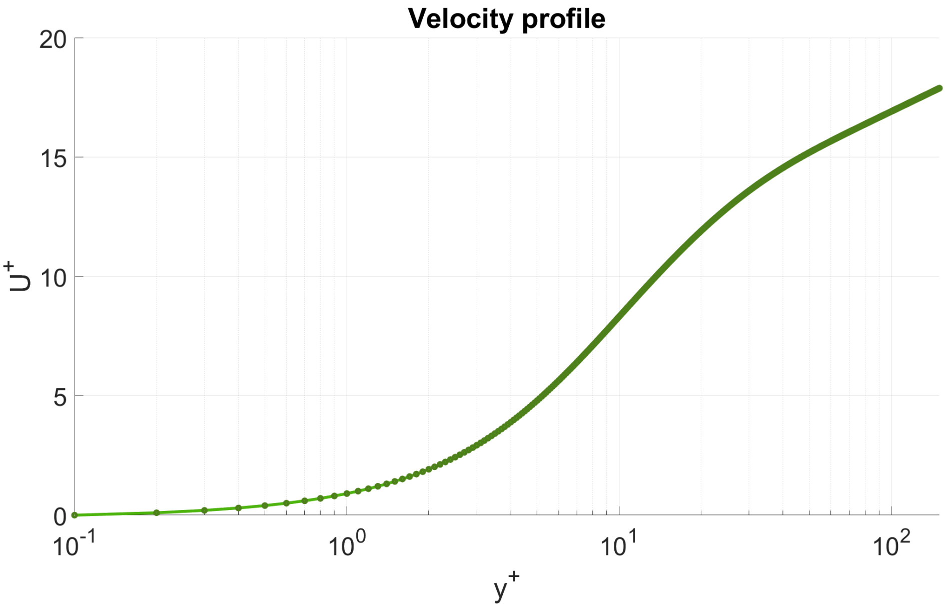


Figure 4.4: Velocity profile inside the channel in function of y^+.

Eulerian Parameter Derivation

The next step is the derivation of all the Eulerian Parameter as seen in Section 4.2.

Langevin Block

In this section it will be necessary to obtain all the particle velocity parameters outlined in Section 4.3.1. Lookup tables from DNS simulations carried out by Marchioli will be applied at this point. Consequently, the functions will require vectorised lookup tables, which supply precise values of σ_x, σ_y, σ_z, $\frac{\partial \sigma_y}{\partial y}$, $\frac{\partial \frac{\overline{u_1 u_2}}{\sigma_1}}{\partial y}$ through an interpolator. Also $d\xi$s values are calculated in this section using a function that obtains a Gaussian distribution by applying two random numbers and logarithmic distributions. The function is assigned a mean of 0 and a variance equal to the square root of dt.

Fluctuation and Motion Derivation

All the parameters derived are now used to obtain the fluid velocity fluctuations, as seen in Section 4.3.1, to then compute the new particle velocity and position.

```fortran
rhs_fvel_x=sqrt(2.0_rp/tau_l)*dcsi_1+duudy*dt/(1.0_rp+Stk)
rhs_fvel_y=sqrt(2.0_rp/tau_l)*dcsi_2+s_p*dsdy*dt/(1.0_rp+Stk)
rhs_fvel_z=sqrt(2.0_rp/tau_l)*dcsi_3
!
if(y_p<=150.0_rp) then
!
    vel_r_x=u_p*u_tau+fvel_x(p)*sigma_x-vel_x(p)
    vel_r_y=          fvel_y(p)*sigma_y-vel_y(p)
    vel_r_z=          fvel_z(p)*sigma_z-vel_z(p)
!
else
!
    vel_r_x=u_p*u_tau-vel_x(p)
    vel_r_y=         -vel_y(p)
    vel_r_z=         -vel_z(p)
!
endif
!
vel_r=sqrt(vel_r_x**2+vel_r_y**2+vel_r_z**2)
tau_p=2.0_rp/9.0_rp*rho_r*rad(p)**2/nu
Re_p=2.0_rp*vel_r*rad(p)/nu
f_p=1.0_rp+0.15_rp*Re_p**0.687_rp
!
rhs_vel_x=f_p*vel_r_x/tau_p
rhs_vel_y=f_p*vel_r_y/tau_p
rhs_vel_z=f_p*vel_r_z/tau_p
!
rhs_pos_x=vel_x(p)
rhs_pos_y=vel_y(p)
rhs_pos_z=vel_z(p)
!
pos_x(p)=pos_x(p)+vel_x(p)*dt
pos_y(p)=pos_y(p)+vel_y(p)*dt
pos_z(p)=pos_z(p)+vel_z(p)*dt
!
vel_x(p)=vel_x(p)+rhs_vel_x*dt
vel_y(p)=vel_y(p)+rhs_vel_y*dt
vel_z(p)=vel_z(p)+rhs_vel_z*dt
```

```
!
if(y_p<=150.0_rp) then
!
    fvel_x(p)=(fvel_x(p)+rhs_fvel_x)/(1.0_rp+dt/tau_l)
    fvel_y(p)=(fvel_y(p)+rhs_fvel_y)/(1.0_rp+dt/tau_l)
    fvel_z(p)=(fvel_z(p)+rhs_fvel_z)/(1.0_rp+dt/tau_l)
!
else
!
    fvel_x(p)=0.0_rp
    fvel_y(p)=0.0_rp
    fvel_z(p)=0.0_rp
!
endif
```

Particle Repositioning

At the end of process, it is necessary to make sure that the particle has not gotten out of the bounds of the simulation domain, and if that happens, boundary conditions will be applied with the use of the function ***ReposPart*** shown here.

```
subroutine ReposPart
    implicit none
    integer :: p
    !
    do p=1,N_P
      !
      if(pos_x(p)<0.0_rp) pos_x(p)=pos_x(p)+L_x
      if(pos_x(p)>L_x   ) pos_x(p)=pos_x(p)-L_x
      !
      if(pos_z(p)<0.0_rp) pos_z(p)=pos_z(p)+L_z
      if(pos_z(p)>L_z   ) pos_z(p)=pos_z(p)-L_z
      !
      if(pos_y(p)<rad(p)) then
        !
        pos_y(p)=2.0_rp*rad(p)-pos_y(p)
        vel_y(p)=-vel_y(p)
        !
      endif
      !
      if(pos_y(p)>L_y-rad(p)) then
        !
        pos_y(p)=2.0_rp*L_y-2.0_rp*rad(p)-pos_y(p)
        vel_y(p)=-vel_y(p)
        !
      endif
      !
    enddo
    !
    return
end subroutine ReposPart
```

4.5 Results

In this section are presented the results obtained by the simulations described in Chapter 4. These were carried out using the Fortran 90 compiler, as well as the post-simulation operation. As for visual simulation, the go to software was Paraview, a open-source application for visualization of scientific data of various nature. The graphs are obtained with the aid of Matlab.

4.5.1 Particle Statistics

When computing particle statistics, it is necessary to define precisely the computational procedure to ensure the reproducibility of the results, but also it is necessary to define a ”grid” on which the statistic will be observed.

Wanting to validate previous data, the same process of Marchioli et al. [16] will be used.

All the statistic are computed over $N_s = 193$ wall-parallel fluid slabs distributed non-uniformly along the wall-normal direction. The thickness of any sth slab, $\Delta y^+(s)$, is obtained via hyperbolic-tangent binning, with a stretching factor γ=1.7:

$$\Delta y^+(s) = \frac{Re_\tau}{tanh(\gamma)}\left(tanh\left(\gamma\frac{s}{N_s}\right) - tanh\left(\gamma\frac{s-1}{N_s}\right)\right) \tag{4.24}$$

This way, the smallest thickness is at the wall, where the largest particle concentration gradient is expected, while the largest thickness is at the channel centerline. The laminar and the buffer layer near the wall, which are the regions where anisotropic effects are strongest, will be well resolved. A particle belongs to a particular slabs when its center is located inside the slab.

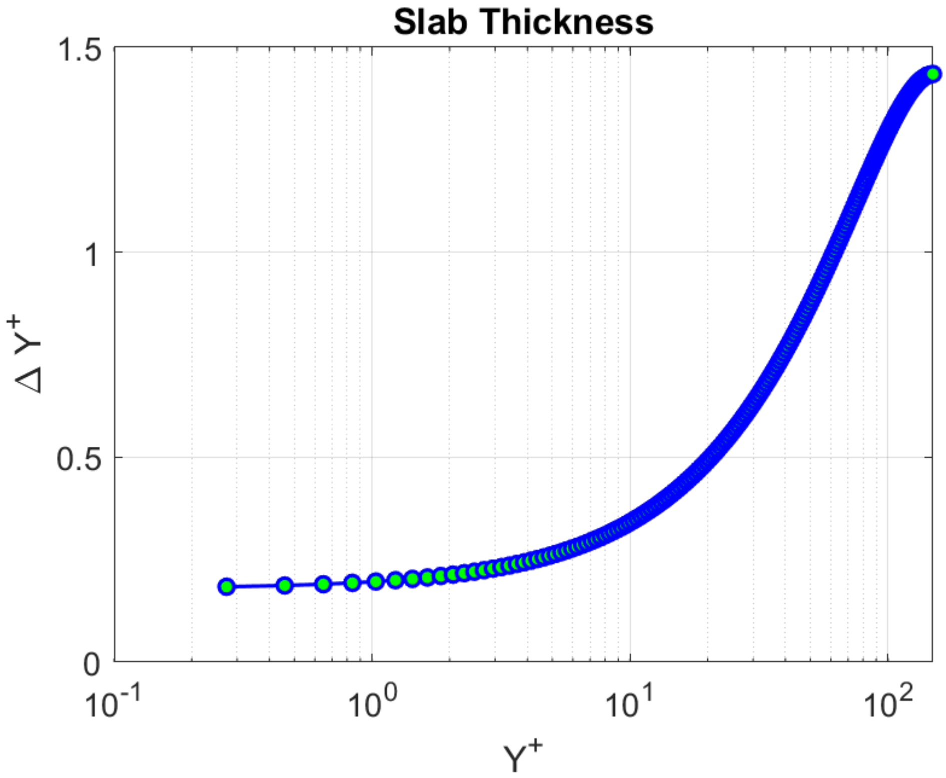


Figure 4.5: Slab thickness with increasing distance from the wall.

The point of the simulation is to reach a statistically steady-state for the particle distribution, so a certain time is needed to allow the concentration close to the wall to get to its steady state, given that its the one that will take longer.

After a significant initial fluctuation lasting approximately 100 wall time units and a subsequent gradual convergence towards a steady state, the system reaches stability at t^+=1,200. Since the non-dimensional bulk velocity in the channel is approximately equal to $U_x^+ = 18$,

this time threshold corresponds to a developing-length of around 1000 channel heights.
This figure highlights the challenges, both in numerical and experimental contexts, of acquiring data on particle-laden channel flows that have developed fully. The length of particle development can exceed the length of hydrodynamic development, thereby necessitating excessively long computational simulation times and extended channels (or pipes) in physical experiments.

4.5.2 Concentration Profiles

The particle concentration profiles are studied in this section in order to better understand the interactions between turbulence and particle inertia. To compare results, the same four class of particles used in Dehbi [8] are considered.

Stk (τ_p^+)	d_p (μm)	d_p^+	$\tau_p(s)$
0,2	$9,59$	$7,2 \cdot 10^{-2}$	$2.4 \cdot 10^{-4}$
5	$48,0$	$0,360$	$5,97 \cdot 10^{-2}$
25	$107,33$	$0,805$	$2,99 \cdot 10^{-2}$
125	$239,99$	$1,8$	$0,149$

Table 4.1: Particle classes and parameters.

In each simulation 25000 particles are generated randomly in the fluid domain: larger samples do not show significant difference in the results.
The data presented compares the results acquired from the thesis simulator with those obtained from Debhi's research and Marchioli's DNS. The objective is to obtain results similar to that of Debhi's rather than to achieve DNS-level accuracy. For each particle class there will be a graph showing the concentrations taken after $\sim$ 35 and $\sim$ 60 seconds, the same time taken by Debhi.

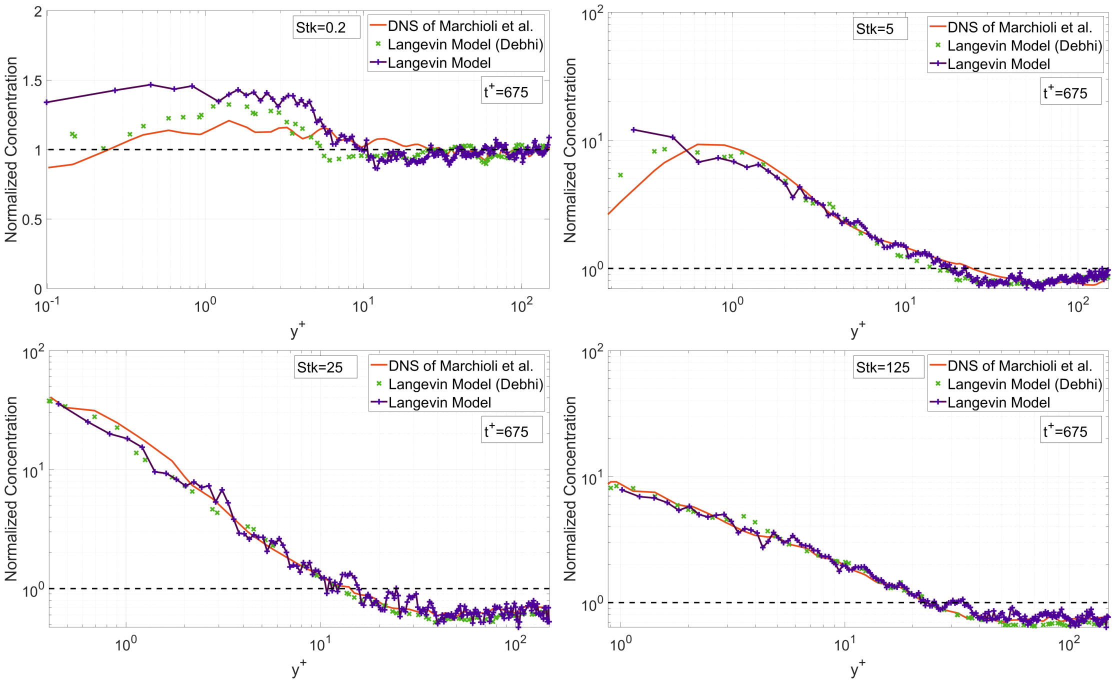


Figure 4.6: Particle Concentration at $t^+ = 675$.

Generally speaking:

- The stochastic Langevin model predictions are in excellent agreement with the DNS results, both in terms of trends and magnitudes. The results from this this work are not perfectly the same as in Dehbi [8], but that can traced back to the fluid motion being simulated from a simple profile instead than from a complete simulation. Also the number of bins is determinant on the exact values obtained. So, looking only at the trends, the result can be considered satisfying;

- Particles with very low inertia tend to remain approximately well mixed in the channel regardless of the time spent;

- Particles with medium and high inertia develop substantial concentration peaks well inside the laminar layer. These peaks increase as time evolves, indicating that steady state concentrations require very long times to be reached;

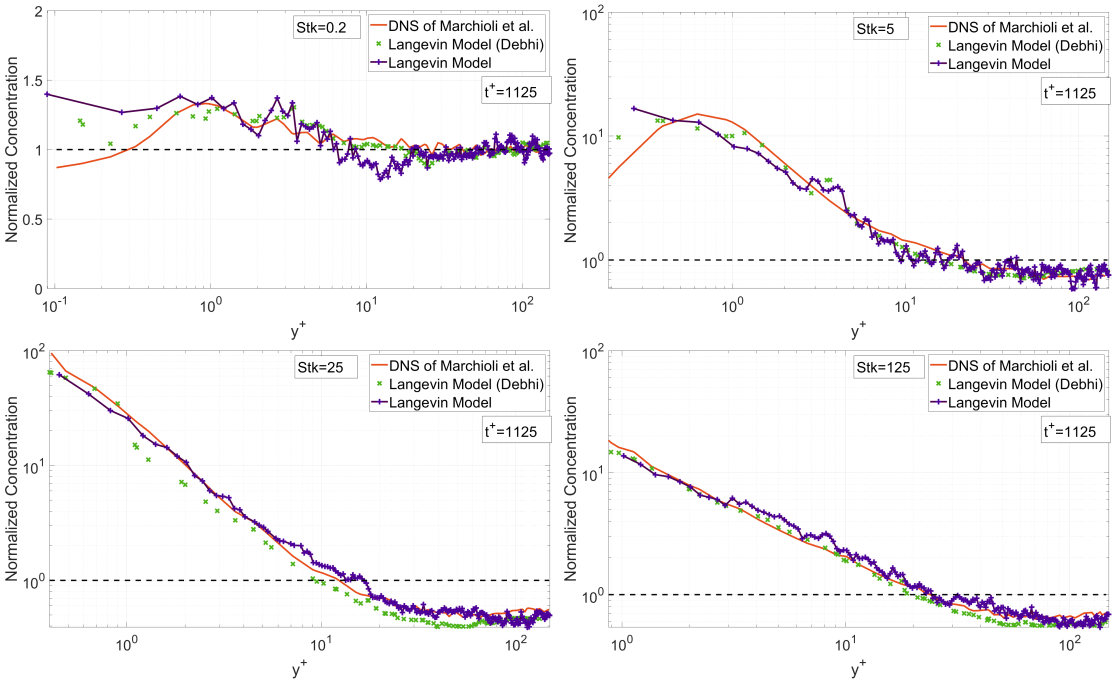


Figure 4.7: Particle Concentration at $t^+ = 1125$.

- The largest peaks are recorded for mid-range inertia particles. These particles are thus the most affected by turbulence. Particles with the highest inertia display smaller peaks because their motion is less affected by fluid fluctuations.

Overall, the obtained concentrations are satisfying. Although the graphs are not a perfect match, Debhi's trends are closely followed. Specifically, the concentrations appear to be slightly more uncertain, which may be attributed to simulating only half as many particles as in Debhi's study or to a smaller timestep being required.

In any way, these uncertainties only become more apparent in the case of Stk=0.2, i.e. particles that are practically comparable to tracers and therefore have a concentration profile that is basically traceable to a white noise profile.

4.5.3 Mean Velocities

The particle velocities are computed using the same process of the data it is compared to: at selected time steps, the bin containing the particle is determined, the instantaneous ensemble-averaged velocity for the bin is computed, and then these mean velocities are averaged over time for a prescribed sample of time steps.

Both for velocity in the axial and in wall-normal direction, the time period over which these quantities are calculated is from $\sim$ 40 seconds to $\sim$ 60 seconds.

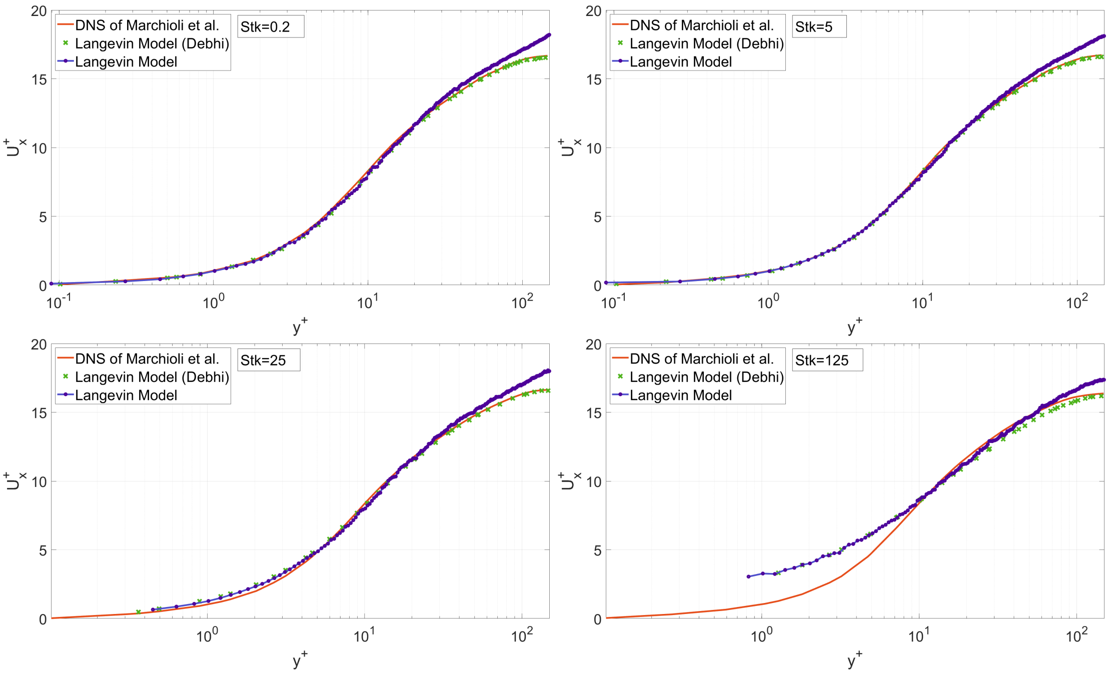


Figure 4.8: Mean axial velocity for the four particles classes.

Looking at the axial velocity, in the original model there where only slight differences between the particle axial velocities and the fluid velocity. The two profiles are essentially identical up to Stk=25. For Stk=125 the particles are faster than the fluid in the laminar sublayer, but slightly lag the fluid in the logarithmic region.

The results from this work are in agreement with this trend. There's only a slight difference in the Stk=125, where the particles lag less then the Dehbi result. The "spike" present after

$y^+ = 100$ is due to the difference between the channel velocities in this thesis work and the ones in the nomenclature. Instead of the data from a DNS, the simpler profile shown in Section 4.4 was used, resulting in a significant difference when looking at this particular statistic. Anyway, this does not result in a problem for the simulation, since particles are very quickly drawn away from the center of the channel.

It is also very important to observe the mean wall-normal particle velocities, because they are direclty linked to the rate of deposition. The normal velocities from the DNS profiles and the Langevin model are compared. The positive normal direction is away from the wall. For Stk=0.2 the wall normal velocity is essentially at noise level. For higher inertia particles, the Langevin model and DNS profiles match almost perfectly. The maximum velocity is largest for particles with St=k25 and decrease for larger particles, as they are bigger and less sensitive to turbulent fluctuations.

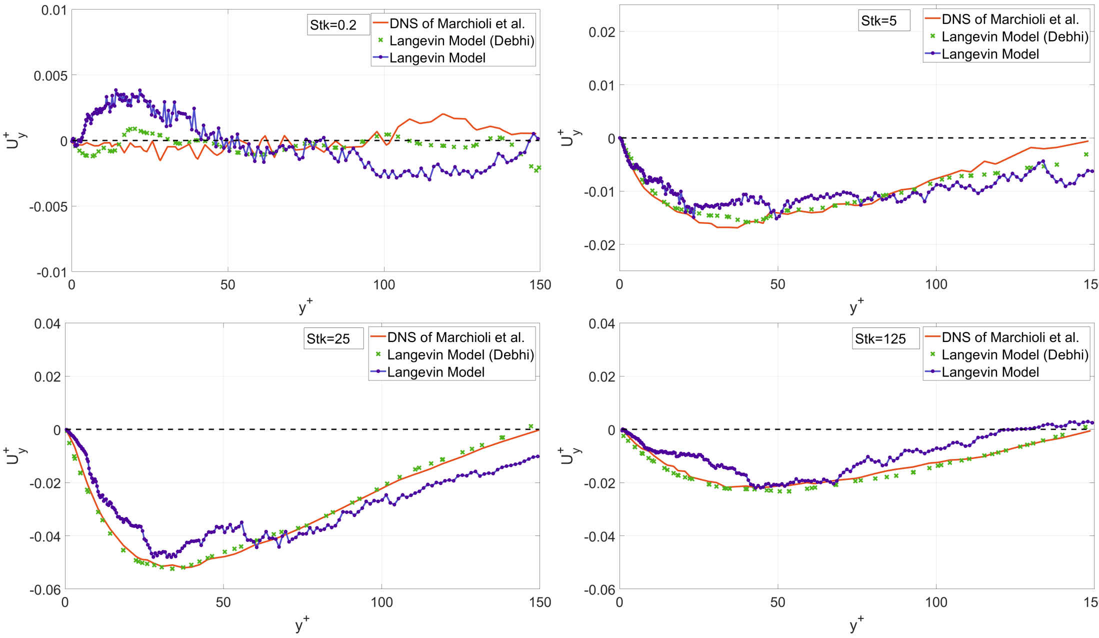


Figure 4.9: Mean wall-normal velocity for the four particles classes.

Focusing on comparing the results of both models, the outcome here is satisfactory despite the challenge in gathering data for this statistic. The magnitude of this parameter is small and requires abundant initial data to ensure convergence of the average velocity. Even a slightly higher timestep can cause the results to be inaccurate. The convergence of statistics necessitates significant data, resulting in a large, challenging simulation. Even minor variations in the time frame can cause significant differences in results.

Nonetheless, the trend is significantly consistent, and the outcomes are mostly coincident. The less precise results are for the Stk=0.2 scenario, but this is because the average velocity magnitude is similar to noise level. As a result, even minor adjustments in the integration time can result in entirely distinct profiles each time.

As far as the z-axis is concerned, nothing is reported, because cyclic conditions are applied in this direction and because Langevin's model in this case only implements random elements, so any result has no specific meaning other than to give an average velocity oscillating around 0.

4.5.4 Rms of Velocities

Another parameter that can be useful when examining the results is the root mean square. Essentially, it provides an insight into the average activity level of particles compared to the mean motion in that segment of the channel. By doing so, it enables us to identify where particles are affected by inhomogeneities, but also where they tend to stop.
The parameter is evaluated in the same time frame in which the mean velocities were calculated.

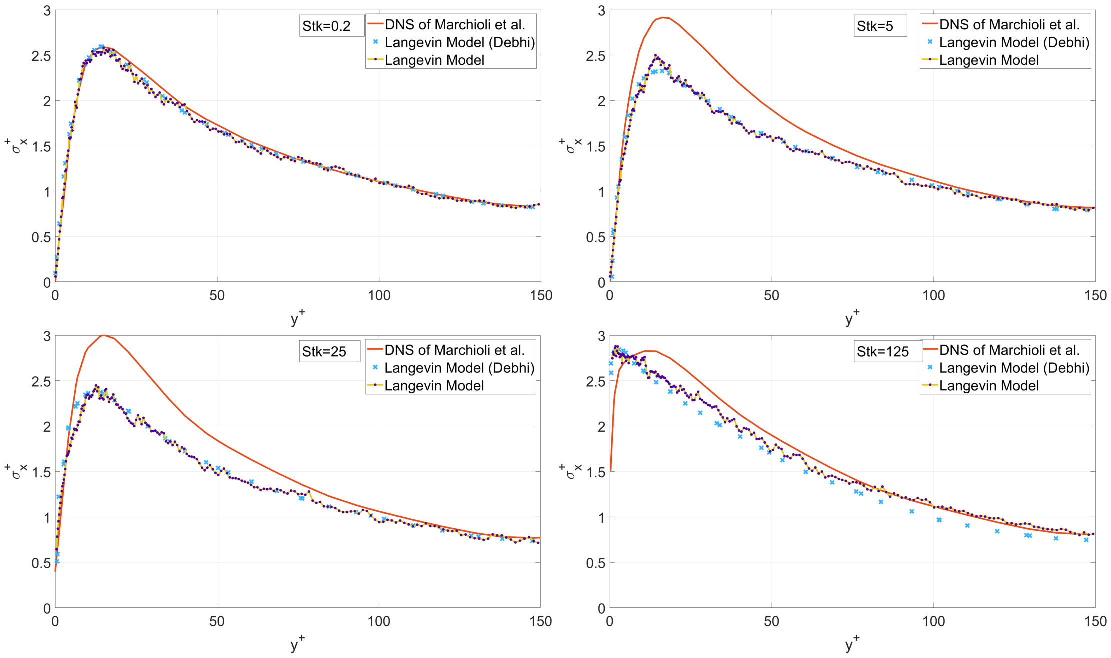


Figure 4.10: Rms of axial velocities.

The thesis results are perfectly consistent with the Langevin ones. Both are quite close to the DNS profiles, but tend to deviate from it as the particle inertia increases, both in heigth and peak position (speaking of the axial velocity).

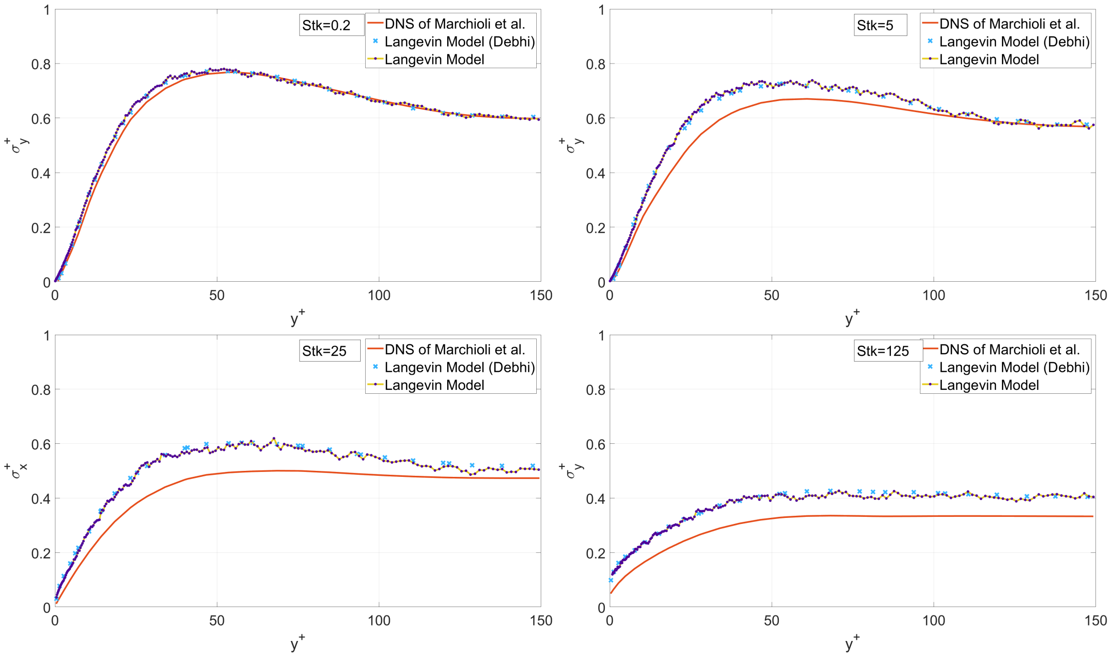


Figure 4.11: Rms of wall-normal velocities.

For the mean wall-component the same can be said, thesis and Debhi are in perfect agreement. Compared to the DNS profile, the more the particle inertia increases, the more the model overpredicts this quantity.

The deviations from the DNS data can be linked to the assumption of Gaussian turbulent scales done in the stochastic part of the Langevin equations.

As in Section 4.5.3, the z-component will not be reported.

4.5.5 Importance of the Stokes parameter in the drift correction term

As discussed previously in this paper, numerous authors have derived the drift correction term A_i for fluid particles in the normalized Langevin equation.

However, particles with sufficient inertia will not experience the same turbulence as fluid particles. Researchers have attempted to address this issue by, for example, implementing the fluid correction while modifying the time scales in the stochastic equation. Although this method gave good results compared to DNS, the simulated particles were not inertial enough to deviate significantly from fluid particles.

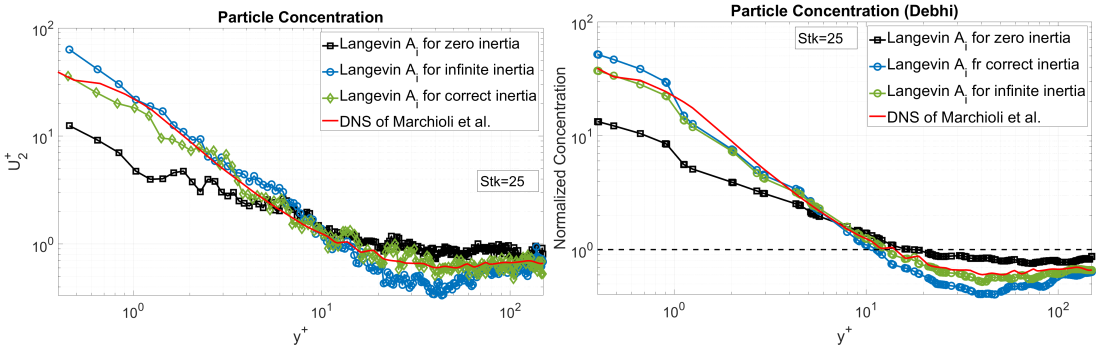
In the studies carried out by Debhi, the good results obtained by the stochastic model over the range of particle inertia are largely due to the fact that the particle inertia is included in the formulation of the drift correction term in the Langevin equation.
As discussed previously, it has been demonstrated that the correction for the drift of a heavy inertial particle can be obtained from the drift correction of a fluid particle, multiplied by a factor that incorporates the particle's Stokes number. This is achieved by analysing the total differential of the fluid velocity fluctuation along the inertial particle's path:

$$A_i = \frac{\partial \frac{\overline{u_j u_i}}{\sigma_i}}{\partial x_j} \cdot \frac{1}{1 + Stk} \tag{4.25}$$

This value is derived by simplifying the equation of motion for particles under locally homogeneous conditions and assuming a typical exponential decorrelation of turbulence that corresponds to the integral time scale of the fluid for a large number of particles.
This equation is accurate at the limits: when the particle has extremely low inertia (Stk=0), the drift correction decreases to the value necessary for a fluid particle, while when the particle has high inertia (Stk»1), the drift correction reduces to zero. Consequently, the particles are not affected by the turbulent fluctuations.

To demonstrate how the non-inclusion of the Stokes number affects the predictions, the same test performed by Debhi was carried out: the same two simulations for Stk=25 were conducted. In the first simulation, the Stokes number was set to 0, whereas in the second simulation, it was set to infinity. In the latter case the factor 1/(1+Stk) is thus equal to 0.
Particle with this inertia possess Stokes numbers that actually fluctuate within the boundary layer, ranging from 0.41 in the outer edge, to 2.65 in the laminar sublayer. Therefore, according to Equation (4.25), the drift correction for these particles should be within the range of 0.27 to 0.71 times the correction required for a fluid particle. To prove what has been said,

just 2 statistics are needed: the predicted concentration profiles after $\sim$ 40 seconds, and the mean normal velocity taken between $\sim$ 35 and $\sim$ 60 seconds.



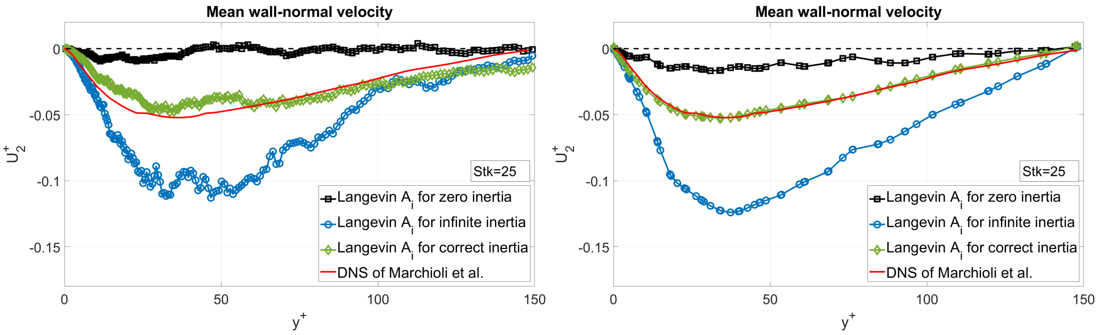
(a) Own Model.

(b) Debhi's Model.

Figure 4.12: Comparison between Debhi and own results.

Once again the results from the thesis simulator and Debhi's are presented. They are overall in agreement, even if one can notice a bit more uncertainty from the data of this work. The explanation resides once again in the timestep and in the different velocity profile applied.

If the Stokes number is set to 0, the computed concentration widely underpredicts the DNS data in the laminar sublayer significantly since particles are kept from the wall by a correction that is too large. The velocity towards the wall is significantly lower compared to the DNS results. However, for large Stokes numbers resulting in zero drift correction, the particle concentration is artificially increased near the wall, causing a substantial increase in the wall-normal velocity compared to the DNS data.



(a) Own Model.

(b) Debhi's Model.

Figure 4.13: Comparison between Debhi and own results.

To achieve precise results for particle dispersion, it is therefore crucial to include the Stokes number effect in the drift correction term of the normalised Langevin equation.
It is worth noting that there is a PDF analysis conducted by Arcen et al. [3], that indicates that there is a decline in the accuracy of the predictions, particularly at high inertia, by disregarding the impact of inertia on particle fluctuations as observed in the latter simulations.

Therefore, it can be concluded that the Langevin model implemented by Debhi has been accurately replicated. In the following chapter, the model developed will be tested to see if it is still valid under other conditions, at what level it is still valid and, if necessary, what corrections need to be made to improve the results obtained.

5

Higher Reynolds Cases

In the previous chapter the model developed by Dehbi, a Continuos Random Walk model based on the Langevin equation, has been implemented and validated through confrontation.

Now that the simulator developed in this thesis has been validated, it is necessary to investigate whether this model remains valid in all circumstances or to what extent it deviates from reality. It should be noted that many of these models have a limited range of applicability, and certainly the Reynolds number at which this model was developed is far from representing the Reynolds number of any realistic atmospheric simulation.

An attempt will therefore be made to implement the same model but at more realistic Reynolds values, making the appropriate modifications to the velocity standard deviations and analytical formulas that the model requires. Therefore in Section 5.1 there will be a show case of these necessary adjustments, in Section 5.2 all the results will be presented, and finally in Section 5.3, the outcome will be commented, evaluating criticities and possible solutions.

5.1 Data Adjustments

During the model implementation, fluid behaviour statistics from a DNS performed externally to this thesis were utilized several times. However, these statistics are only applicable to a simulation where the fluid has a $Re_\tau = 150$, and altering this property results in inconsistent statistics for the new simulation. In this study, there was no DNS of its own, so it was necessary to find additional data to that of the original model, in order to have a source for the necessary statistics, such as the σ_i's of the velocities, but also results on the concentration distributions to compare with.

This kind of data was found in the work of Bernardini et al. [5], who conducted a study on velocity statistics in a turbulent channel flow, with a Re_τ of up to 4000. Bernardini et al.'s research shares the general objective presented at the beginning of this thesis, i.e. to extend the knowledge of turbulent flows at high Reynolds numbers, but here it is limited to the study of fluid properties without the addition of inertial particles. This is not an issue since the simulation results still offer the necessary statistical data to update the model currently being implemented.

To assess the situation , the elements that need to be updated or checked are: the profile of the flow velocity in the channel, the flow velocity variances, and the quantity τ_l.
They require re-evaluation due to their derivation from a fit based on DNS data or their dependence on the y^+ wall coordinate, which has the same magnitude Re_τ, as explained in Chapter 2. Therefore in this new simulation, it will also be increased requiring new fits or adaptations.

Fluid Velocity Profile

As presented in Section 3.3 the fluid velocity profile is obtained from:

$$\frac{u}{u_\tau} = \frac{1}{k} log(1 + ky^+) + C_k \left(1 - e^{-y^+/11} - \frac{y^+}{11} e^{-0.33y^+}\right) \tag{5.1}$$

This profile, can be considered still valid at higher Reynolds, as the log distribution in lower coordinates is still respected, as well as the linear region [5]. The obtained profile is presented in Figure 5.1.

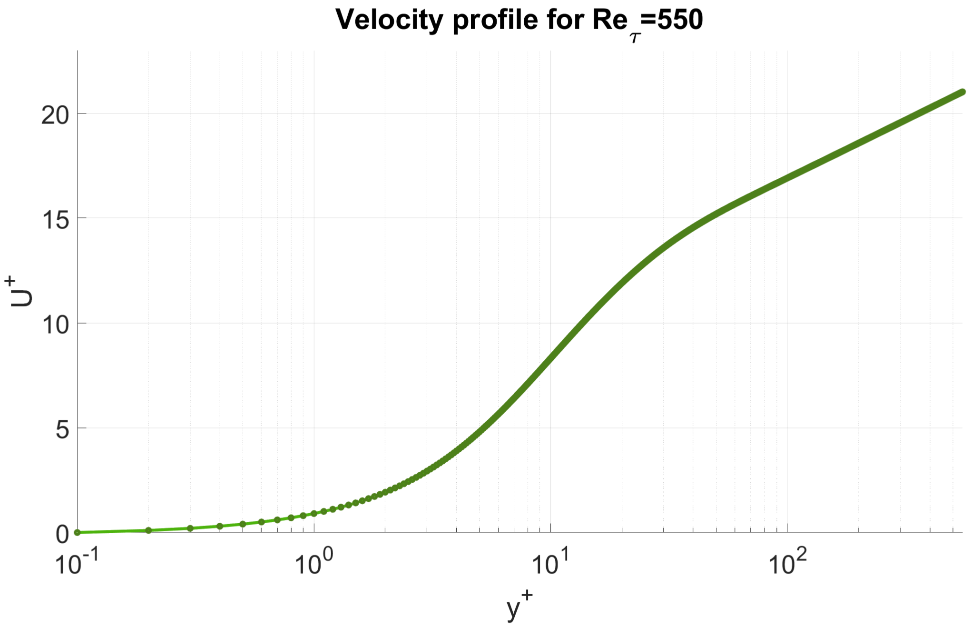


Figure 5.1: Velocity profile for $Re_\tau = 550$.

Fluid velocity deviations

Contrary to the speed profile, this parameter undergoes a strong modification as the Re_τ increases. In fact, as the Reynolds number increases, so does the generic Reynolds number, causing a chain of changes in the radius of the particle, since all other properties are considered dimensionless or, more importantly, constant: the flow velocity, the materials and consequently the density remain constant, leaving only the size of the particle to vary.
This alters the particle's interaction with the fluid and modifies the degree of deviation from the mean particle velocity. This can be clearly seen in these graphs.

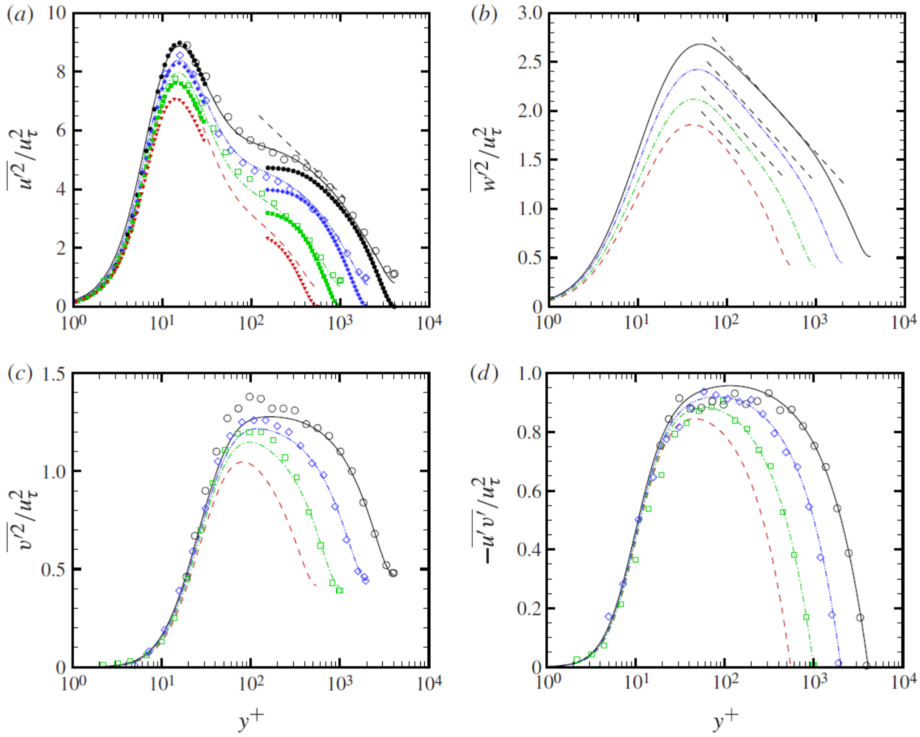


Figure 5.2: The four statistics needed for the model: a)σ_x b)σ_z c)σ_y d)Reynolds stresses, for various Re_τ: red=550, green=1010, blue=1956, black=4000.

These changes will directly affect the computation off the Langevin equations required to compute the velocity fluctuations that are added to mean velocity profile.

Lagrangian Fluid Time Scale

This is perhaps the most complicated issue to deal with. In Debhi's discussion, τ_L is defined in terms of the Lagrangian integral time scale T_L. To obtain these quantities some assumptions are made, that may no longer hold true for higher Reynolds number:

- The Lagrangian integral time scale T_L is notoriously challenging to measure. There is empirical data that indicates a proportional correlation between Eulerian and Lagrangian time scale of the type: $T_L = \beta T_E$, the same relation seen in Section 4.2. So to simplify the process, T_L can be estimated from Eulerian statistics [13]. This assumption is proven valid for low Reynolds and y^+ values, so a different β factor being necessary for higher Reynolds is not excluded, but also finding out if that is true is out of the capacities of the tools being used in this work.

- To obtain the T_L values, Debhi utilizes fits obtained by Kallio and Reeks [13]. These fits are quite well accurate for lower Reynolds, but are obtained from quite old data and also are derived from dissipation rate data that were considered limited at the time already.

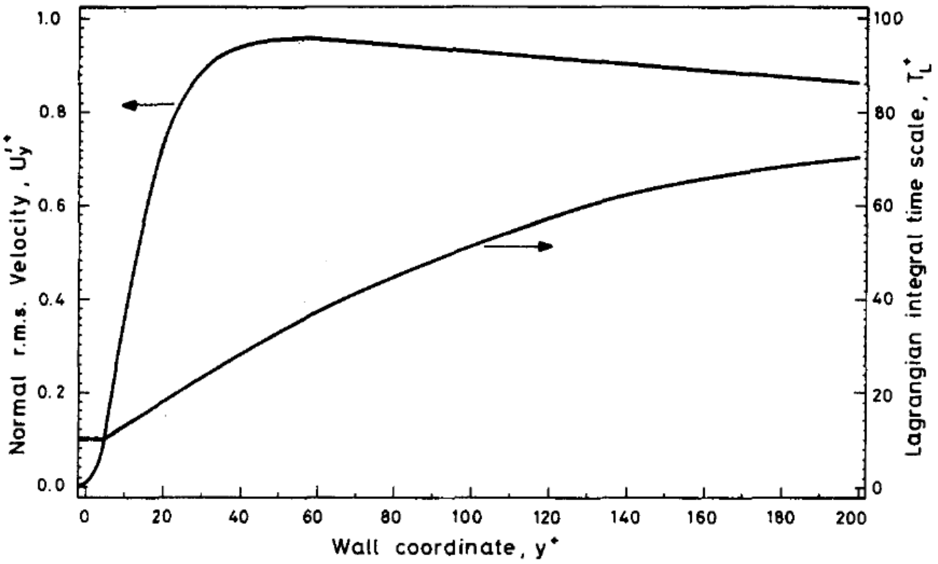
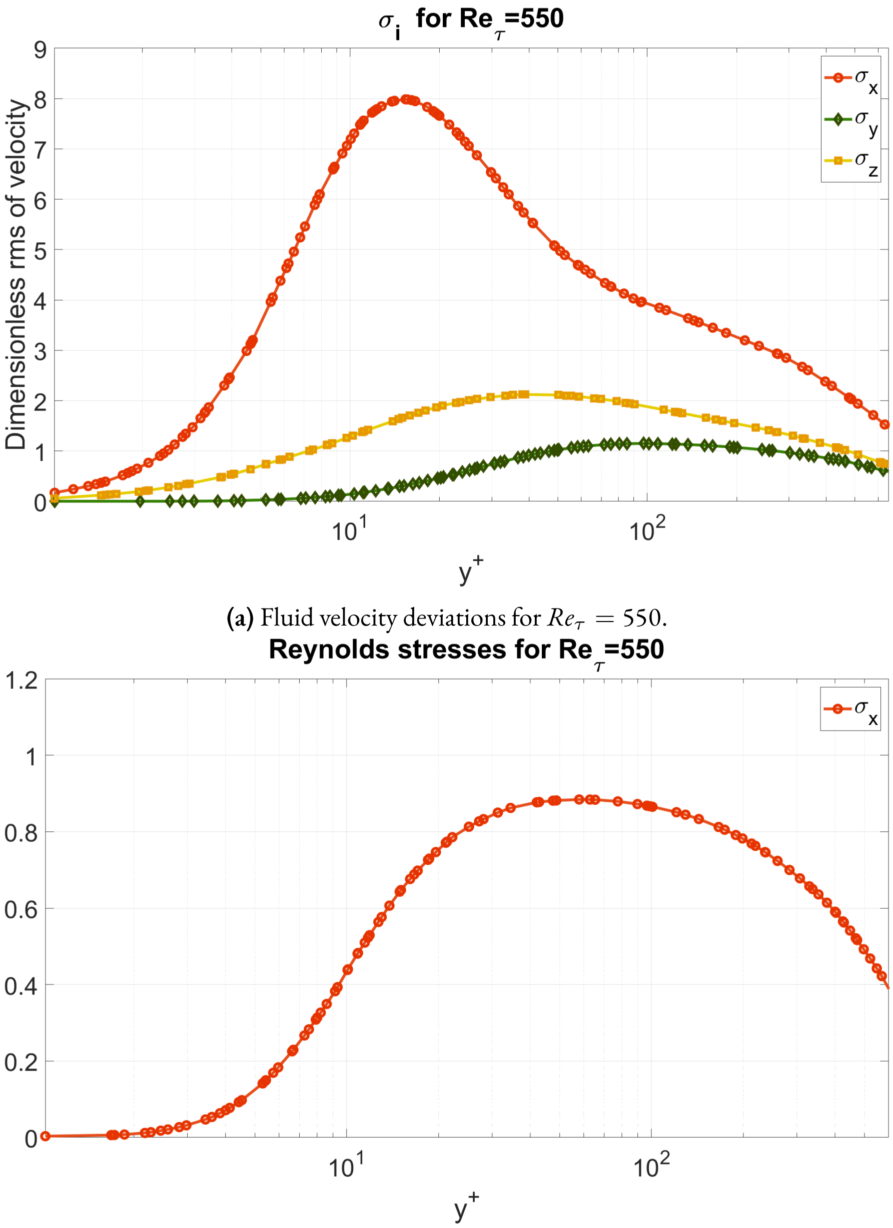


Figure 5.3: Fits for the wall normal rms fluid velocity and the Lagrangian integral time scale[13].

- τ_L is considered equivalent in the 3 directions, but that's actually true only at low Reynolds. Over a certain value, these 3 timescales show a marked difference, that can prove problematic in the simulation.

As a result of the problems presented, the changes made are as follows. The new Re_τ at which the simulations will be performed is 550.

First of all the fluid velocity deviations are updated with the data extracted from Bernardini et al. [5], shown here.



(a) Fluid velocity deviations for $Re_\tau = 550$.

(b) Reynolds stresses for Re_τ.

All the parameters required for the CRW model can be derived directly or by some mathematical derivation from the graphs provided.

It is evident, particularly from the velocity fluctuations, that considerably greater values exist than those for $Re_\tau = 150$.

This can be attributed to the flow becoming more turbulent as Reynolds increases, leading to larger vortex phenomena, especially in the viscous sub-layer. The peaks in the various deviations occur exactly in this region.

These peaks represent the areas within the flow where there is significant variation in the velocity of individual particles compared to the average flow velocity due to the predominance of vortex structures.

In regard to τ_L, the new data is derived from the value T_l interpolated from Figure 5.5. This data is from a DNS way more recent than the fits used by Debhi.

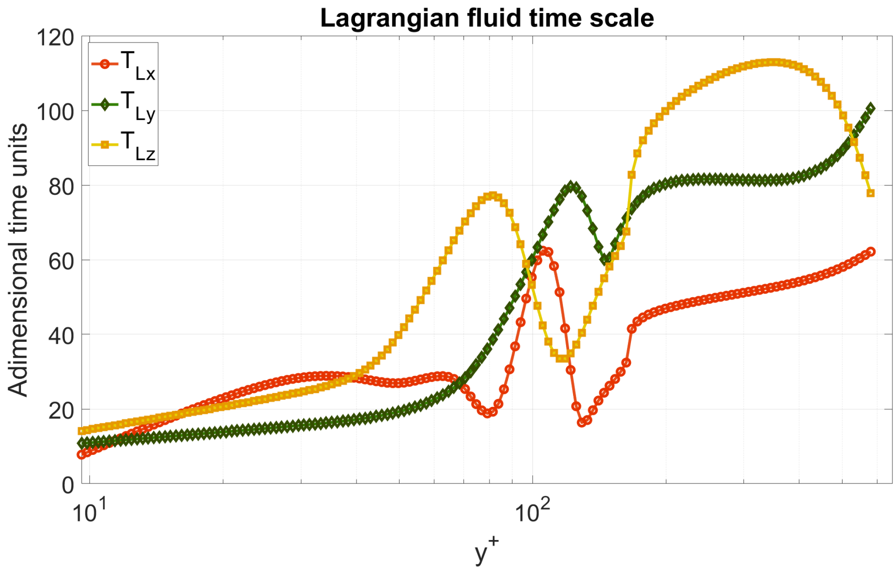


Figure 5.5: Lagrangian integral time scale in wall time units[18].

As one can see, the three values are pretty similar to each other for low y^+ values, and can also be compared to the fits of Kallio and Reeks [13], but from a certain point onward, the profiles become erratic with respect to each other, the most critical point being $y^+ = 100$, the region where typically there is a transition, from describing the velocity profile with the logarithmic law to the outer region, which has a much more linear profile.

To obtain τ_L, different calculations are now required for the three axes, as each axis needs a separate lookup table. This makes the code more laborious, and also more slow in general.

```
if(y_p<10.0_rp) then
    !
    T_lpx=10.0_rp
    T_lpy=11.0_rp
    T_lpz=14.0_rp
    !
elseif((y_p>=10.0_rp).and.(y_p<550.0_rp)) then
    !
    call CmptT_lx(y_p,T_lpx)
    call CmptT_ly(y_p,T_lpy)
    call CmptT_lz(y_p,T_lpz)
    !
endif
!
T_lx=T_lpx*nu/u_tau**2.0_rp
T_ly=T_lpy*nu/u_tau**2.0_rp
T_lz=T_lpz*nu/u_tau**2.0_rp
!
T_ex=T_lx/beta
T_ey=T_ly/beta
T_ez=T_lz/beta
!
tau_lx=T_lx/beta*(1.0_rp-(1.0_rp-beta)\\
*(1.0_rp+tau_p/T_ex)**(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ex)))
tau_ly=T_ly/beta*(1.0_rp-(1.0_rp-beta)\\
*(1.0_rp+tau_p/T_ey)**(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ey)))
tau_lz=T_lz/beta*(1.0_rp-(1.0_rp-beta)\\
*(1.0_rp+tau_p/T_ez)**(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ez)))
```

This concludes the modifications required for the simulator.

5.2 Results

The results obtained under these new conditions will be compared with those obtained by Bernardini in a subsequent study to the one presented before, that further investigates the behaviour of inertial particles in a turbulent channel with a Re_τ of up to 4000[4].

Bernardini's simulation method differs slightly from Debhi's, requiring minor adjustments to the initial conditions:

- The one that has the most impact on the simulation is that in Bernardini's simulations the subject is sand suspended in air, as opposed to water particles. This alteration increases the density of the suspended body to $3500 kg/m^3$, which subsequently alters the density ratio and, in turn, influences particle behaviour. This modification involuntarily provides an avenue for assessing if the model can also accommodate this variation.

- In Bernardini's work the slab division of the domain is uniform instead of incremental. The number of slabs is set at 450. To obtain a similar distribution, the same method is applied here.

- Contrary to Debhi, Bernardini doesn't state after how much time the statistics are taken. Therefore the concentration statistics will here be taken when steady state conditions are obtained for the smaller class of particles, while for the velocity the data will be taken at circa half of that time.

Concentration, mean velocity and velocity deviations data will be presented for 3 types of particles: Stk=25, Stk=100 and Stk=500.
This increase is made both to enable comparison with available data and due to the current higher particle density and also the new bulk Reynolds number being now 20000. Given the same velocity in the channel, the particle size is drastically reduced for the same Stokes number. To avoid computational issues, it is necessary to increase the Stokes number.
In addition, there will be a comparison between the two simulations at different Re_τ for Stk=25, since there is available data.

Particle Statistics

Each graph shows the comparison between the thesis model and the DNS of Bernardini[4].

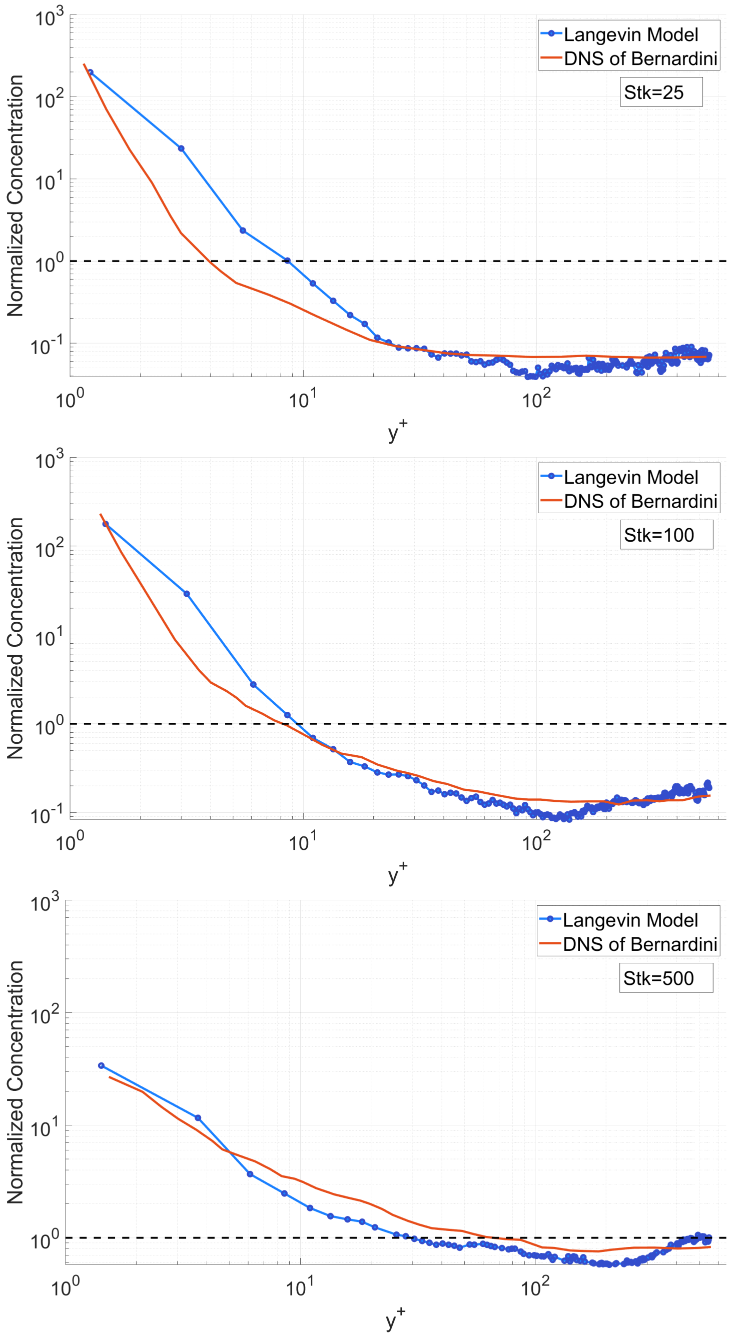
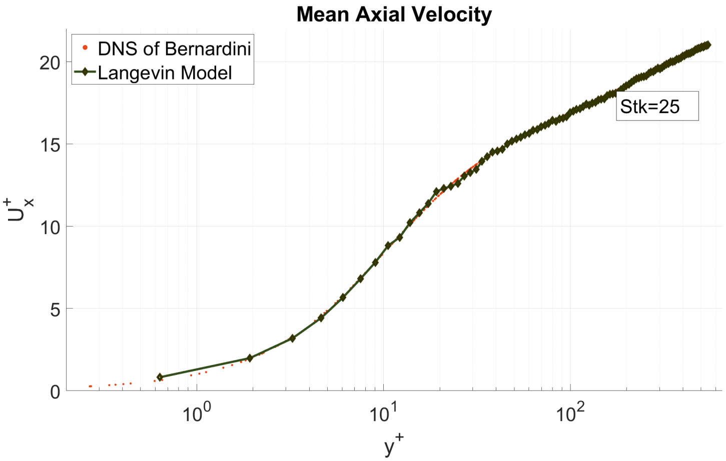


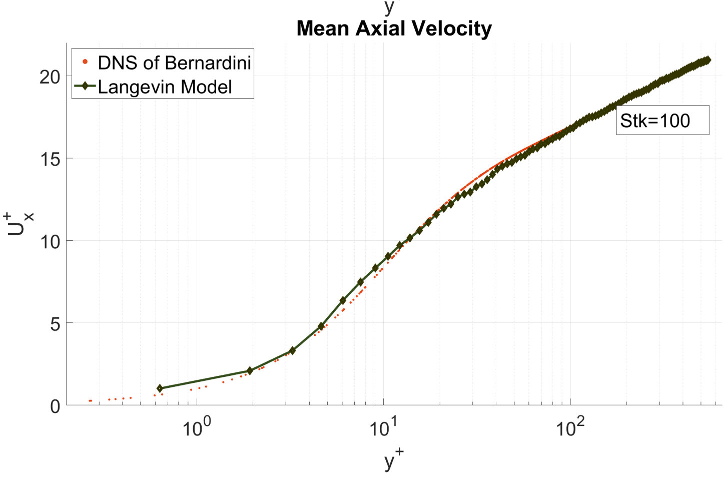
Figure 5.6: Particle concentration for three classes of particles.

What can be seen is that results are not as precise as in the previous case. The trend anyway is still pretty much replicated correctly, with concentration reducing with the increase of the Stokes number. The model seems not capable of replicating the massive peak of concentration at the wall, presenting around 10-15% less particles in the peak, and also having a less steep slope. Things are different for Stk=500, where the concentration is overpredicted instead, by around the same percentage.

Velocity statistics

For these statistics no results are provided from Bernardini [4], so instead a comparison with the velocity profile is applied.





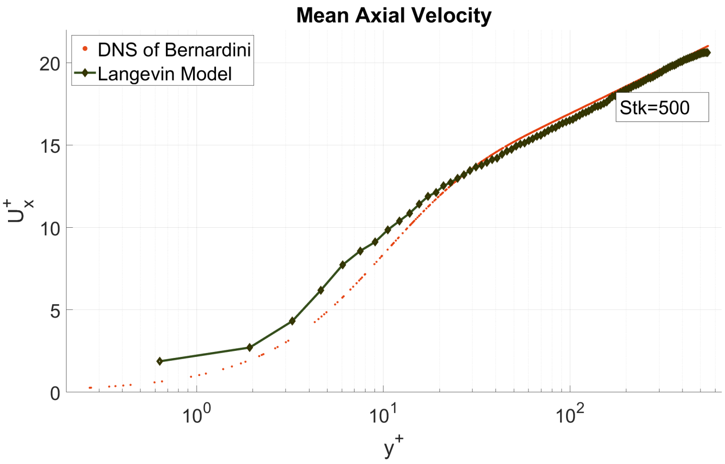


Figure 5.8: Mean Axial Velocities for the three classes of particles.

It can be seen that the particles pretty much respect the velocity imposed by the profile. It also can be seen that as Stokes increases, particle start to lag ahead of the profile, same as what was happening for $Re_\tau = 150$.

As for the wall-normal velocity there is nothing to compare it to, so a comparison between all the Stokes cases will be shown.

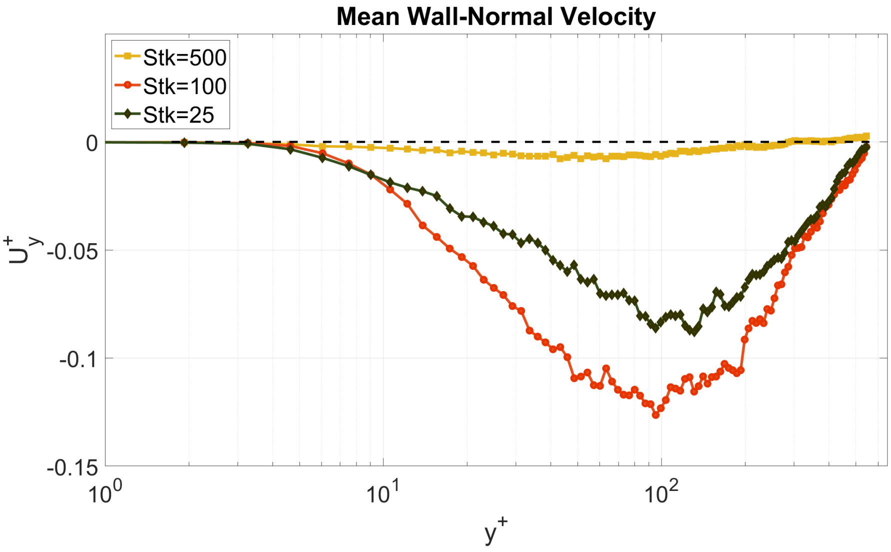
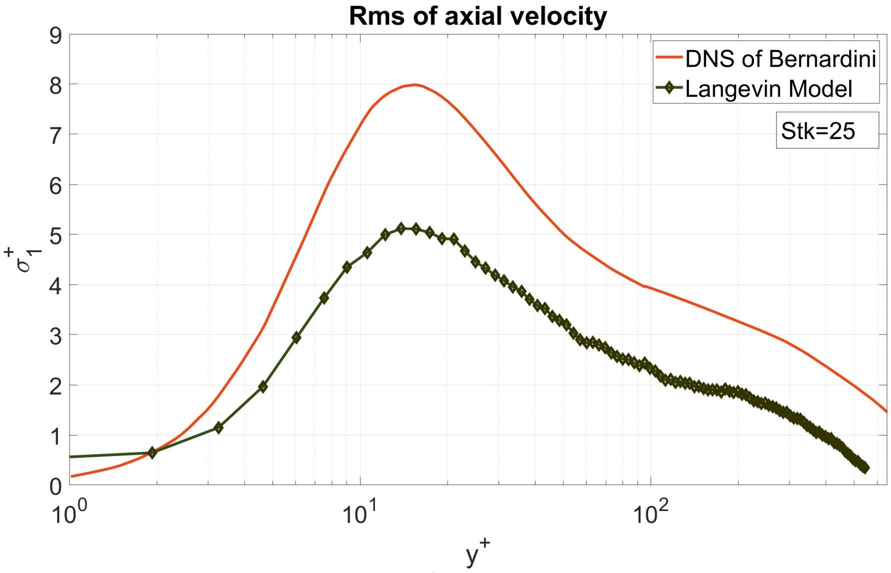
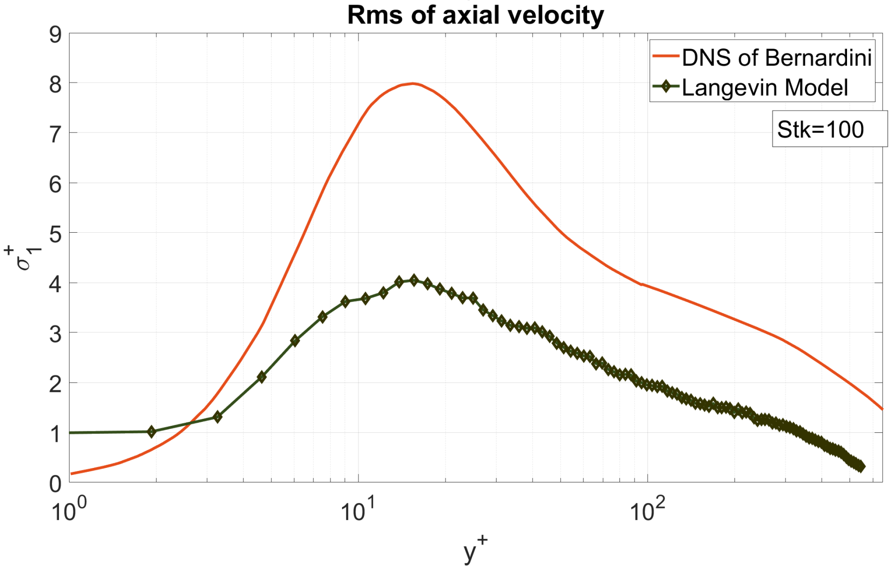


Figure 5.9: Mean Wall-Normal Velocities Comparison.

It is clear that the more the particles grow bigger, the least they are influenced to move perpendicularly to the wall, in agreement with what was shown in Chapter 4.

To conclude the particle velocity analysis, it remains to look at the deviations both in the x and y directions.





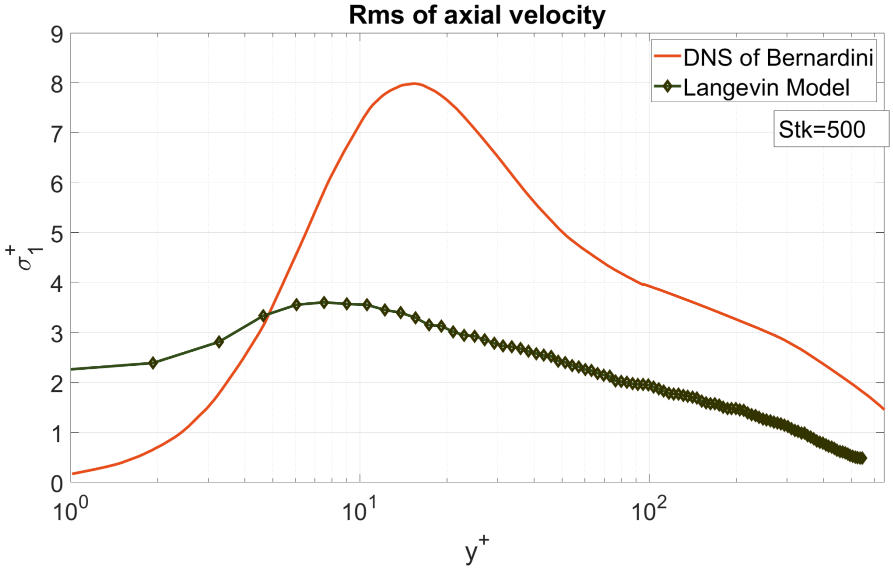
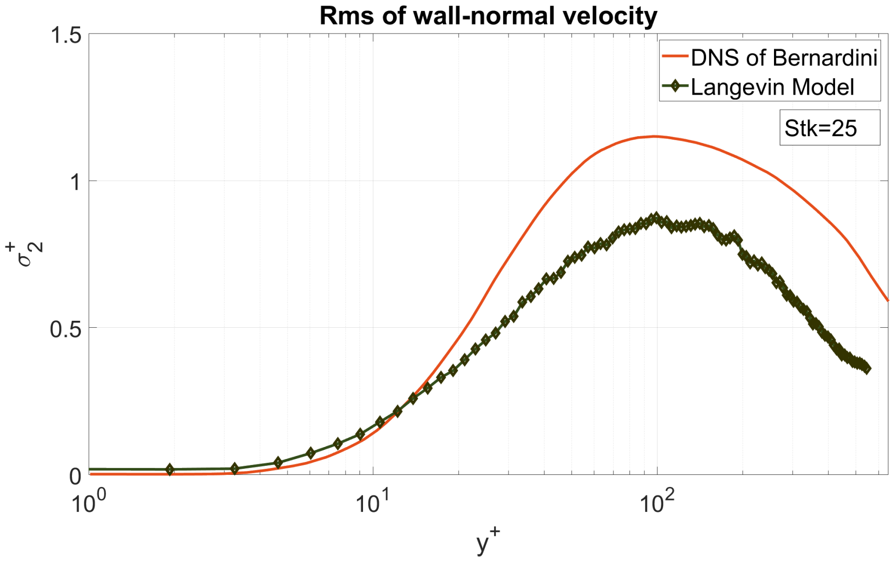
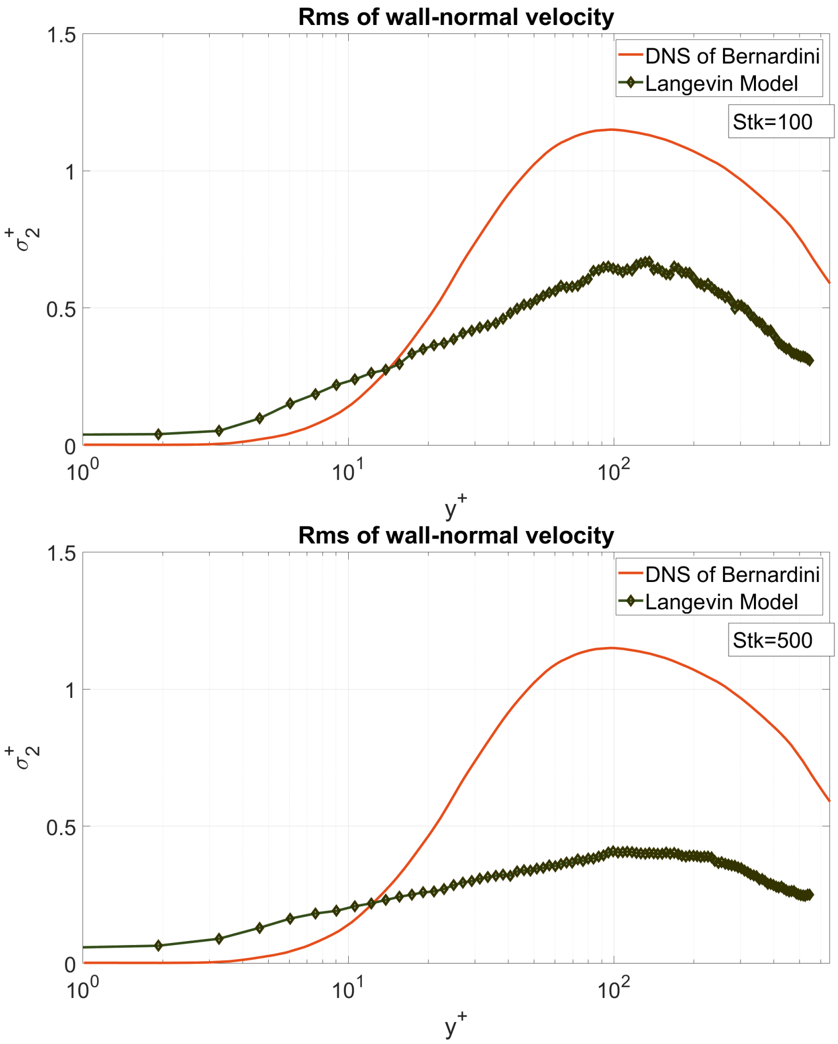


Figure 5.11: Deviations for the Axial Velocity.

In the x direction an expected behaviour can be seen: the more the Stokes number increases, the more the σ_x of the particle can not follow the fluid one. The peak gets smaller and smaller, starting already only at $\sim$ 5 for Stk=25, as opposed to the fluid $\sim$ 8. It can also be noticed a slight shift to the left of he whole statistic.





(a) Deviations for Wall-Normal Velocities.

The pattern here is the same as in the axial velocity, with the difference that the shift to the left is less pronounced.

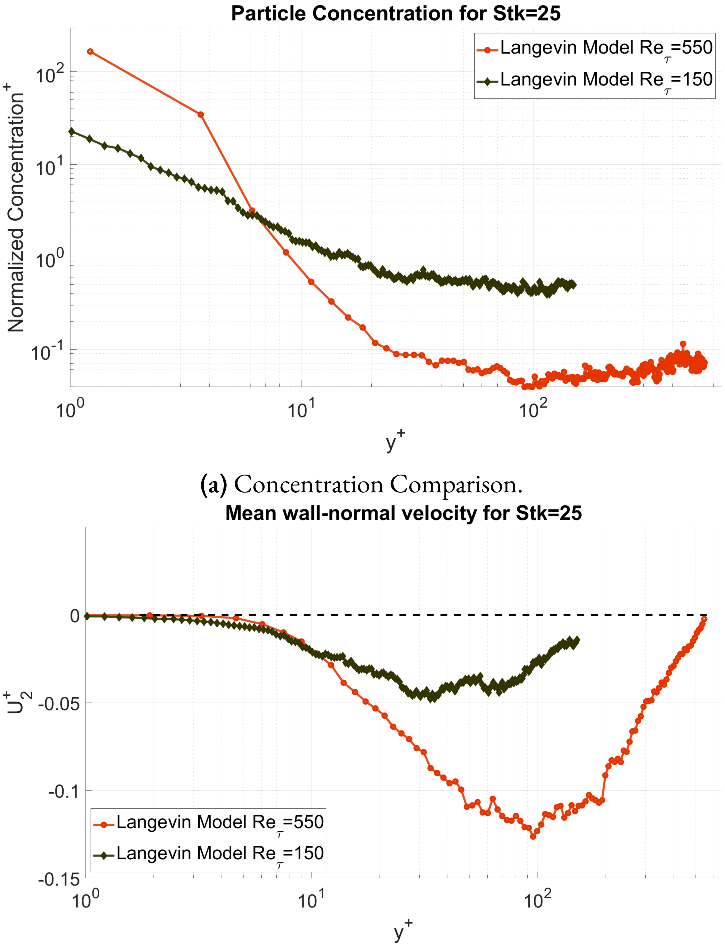
Overall the behaviour seen in these graphs indicates that with the increase of the size of the particle, the mobility respect the mean flow gets smaller and smaller, and the particles are less influenced from the flow in the zone where there should be more vortexes, represented with the peaks in the graphs.

From $Re_\tau = 150$ to $Re_\tau = 550$

A further analysis required is to compare the behaviour of particles with identical Stokes numbers but at various $Re\mathcal{T}$ values.

What is expected is that as $Re\mathcal{T}$ increases, the turbophoretic phenomen becomes more and more pronounced.

A comparison for the particle concentration and the mean wall-normal velocity at Stk=25 will be shown.



(a) Concentration Comparison.

(b) Wall-Normal Velocity Comparison.

Even if the data was collected in different ways for the analysis in Chapter 4 and Chapter 5,

the expected pattern is evident.
It is clear how the particle segregation is increased, going from a normalized concentration of $\sim$ 60 to $\sim$ 180. Also for the wall-normal velocity that is true, with the peak for $Re_\tau = 550$ being $\sim$ 2.5 times bigger than the one for $Re_\tau = 150$.

5.3 Comments and issues

The purpose of this second part of the thesis was to evaluate the model's validity at higher Reynolds numbers.
The theory predicts a universal behaviour for the turbophoretic phenomenon as the Reynolds number of the flow increases. For any given Re_τ number, there is a phenomenon of particle segregation in the wall region of the channel. This phenomenon is more prevalent in particles with intermediate Stokes number values (between 10 and 1000), which have a τ_p of the same order of magnitude as the time scale of the vortex structures present in the buffer layer[4].
This is exactly the behaviour that was achieved with the implementation of this CRW model. Still, it is clear that the results shown do not replicate exactly the DNS of Bernardini.
This issue can initially be attributed to the same factors that prevented Debhi's model from accurately replicating DNS simulations - namely, its dependence on a correction factor that is essentially 'arbitrary', and many other simplifications.
Many other factor in the work of this thesis can produce imprecisions, that once added up, can produce an error on the particle concentration of up to 20%:

- The use of lookup tables is a crucial element in this thesis. The study did not employ its own DNS, but instead obtained some required data by interpolating graphs based on pre-existing DNS data as described earlier. This approach may introduce errors because of inaccuracies, which may be magnified if the data undergoes intermediate derivations or if the values are very small (e.g. Reynolds stresses or the derivative of σ_y).

- In Section 4.2 a series of formulas was presented in the process of the derivation of some required parameters for the implementation of this CRW model. In the switch

to a higher Re_τ various adaptations were presented for these formulas.
Unfortunately, one of the formulas could not be updated.

$$\tau_L = \frac{T_L}{\beta}\left(1-(1-\beta)\left(1+\frac{\tau_p}{T_E}\right)^{-0.4\left(1+0.01\frac{\tau_p}{T_E}\right)}\right) \tag{5.2}$$

Its validity is uncertain as it appears ad hoc and has a constant β that could need adjusting when altering the Reynolds value.

- It is also possible to speculate about the correction factor used by Debhi for the Langevin equations.
  An equation related to fluid fluctuations contains an arbitrary element, incorporated to accurately portray the particle's behaviour based on its inertia. This approach has proven effective for the Stokes ranges examined by Debhi, but is surely a pretty singular one.
  However, an issue could be the fact that the correction factor in question has the Stokes number at its denominator. As detailed in Section 4.5.5, this results in a cancellation of the term and an overestimation of particle segregation when the Stokes number is assumed infinite. It can be speculated that a similar phenomenon occurs with high Stokes numbers, such as those observed in this thesis for Stokes=500. The large inertia is not enough to counterbalance the nearly zeroed correction factor, therefore the predicted concentration is overstimated to the actual one.
  However, this hypothesis lacks specific data or analysis to support it, as it was not within the original scope of the study. Thus there is no intention to present as a valid argument.

6 Conclusions

This dissertation aimed to replicate and then extend an existing stochastic model for the simulation of inertial particles in turbulent flows.

A Lagrangian continuous random walk (CRW) model is used to predict particle dispersion in a channel flow with anisotropic, inhomogeneous turbulence in the wall normal direction. The initial particle tracking model employs 3D mean flow data from the CFD code and Eulerian statistics from DNS databases. The single variation in the thesis model is that the CFD code is replaced by a simpler mean velocity profile.

The normalized Langevin equation is used to obtain the time-dependent fluid velocity fluctuations, which is better equipped for addressing turbulence inhomogeneities. The CRW model includes a correction for drift velocity in random inertia particles.

The model predictions are compared to the DNS data of Marchioli[16], who produced detailed statistics of velocity and transfer rates for four classes of particles having Stokes numbers of 0.2 to 125 and dispersed in a parallel channel flow with $Re_\tau = 150$.

The model is in very good agreement with the DNS data for the various measures of dispersion parameters, i.e.: instantaneous particle concentration profiles and mean and rms profiles

of streamwise and wall-normal particle velocities. It also reproduces well established results such as the build-up of particles in the laminar sublayer, as well as the gradual de-correlation between particles and turbulence with increasing inertia.
In addition, the 'well mixed' criterion is achieved so that tracer particles maintain approximately uniform concentrations when uniformly introduced into the domain and their deposition velocity is vanishingly small.
All of this applies to both the original model and the one implemented in this study.

The second half of the thesis was devoted to the application of the model to higher Reynolds numbers, a test that can be very severe for many models.
A satisfactory result is achieved by comparing the measurements with the data obtained from Bernardini's DNS, specifically with three categories of particles having Stokes number 25, 100, and 500. The expected behaviour is verified thoroughly – the turbo-phoretic phenomenon is universal, as Reynolds number varies, replicating all behaviour observed in the original model application.
However, it is generally observed that some results may be underestimated, such as the concentration near the wall, which is seen to fluctuate up to values of 10-20%.
Based on this investigation, along with previous studies, it is advocated that the normalized Langevin equation should be used to predict particle dispersion accurately in general inhomogeneous flows when the mean field and turbulence statistics are computed or prescribed with a high degree of confidence.

It is clear that the turbophoretic phenomen can be accurately replicated with this model. However, achieving results that mirror reality requires extremely precise data about the mean field and turbulent statistics.

Future and improvements

All the results inaccuracies shown in this work can be attributed to imprecisions in the collection of the necessary data for the model.

To avoid this issue, one possible solution is to create a custom DNS that can provide all the necessary information with the required level of accuracy instead of relying on external data. However, some may argue that this approach would contradict the overall purpose of the model.
To prevent this inconsistency, a more faithful improvement could be simply to increase the quality of the used data. It would be beneficial to have access to highly accurate data on flow and turbulence rather than having to digitise imprecise graphs from prior studies.
Additionally, a more thorough study of some of the analytical formulas used to calculate the fluid Lagrangian time scale could be advantageous.

Still, overall, this type of model and general approach to the investigation of particle motion in turbulent flows has been shown to be effective and efficient. Therefore, a development that focuses on more detailed premises could be a viable option for advancing the study of particle-laden turbulent flows.

7 Appendix

7.1 Integral Version of the Code

To give complete access to the work done in this thesis, the integral version of the code will be reported. The functions defined as lookup tables however will be represented by just an example, that will not present the vectors containing the data derived from the graphs about the σs.

7.1.1 Re_τ=150 Version

```
program ParticleWallModel
  !
  use,intrinsic :: iso_fortran_env,only: rp  => real64
  !
  !======================================================================
  !Parameters Definition
  !
  implicit none
  !
  integer,parameter :: N_P=25000
  integer,parameter :: stampa=100
  integer,parameter :: schermo=10000
```

```
!
real(rp),parameter :: L_x=6.0_rp
real(rp),parameter :: L_y=1.0_rp
real(rp),parameter :: L_z=3.0_rp
!
real(rp),parameter :: rey=4583.44222032_rp
real(rp),parameter :: rho_p=1000.0_rp
real(rp),parameter :: rho_f=1.3_rp
real(rp),parameter :: rho_r=rho_p/rho_f
!
real(rp),parameter :: St=25.0_rp
!
real(rp),parameter :: beta=0.356_rp
real(rp),parameter :: vk_c=0.41_rp
real(rp),parameter :: C_k=7.8_rp
real(rp),parameter :: nu=1.0_rp/rey
real(rp),parameter :: mu=nu*1.2_rp
!
real(rp),parameter :: rey_tau=0.09_rp*rey**0.88_rp
real(rp),parameter :: u_tau=rey_tau*nu/(L_y/2.0_rp)
real(rp),parameter :: l_tau=nu/u_tau
real(rp),parameter :: radius=sqrt(9.0_rp/2.0_rp/rho_r*nu**2/u_tau**2*St)
!
real(rp),parameter :: pi=acos(-1.0_rp)
!
real(rp),dimension(1:N_P) :: pos_x,pos_y,pos_z
real(rp),dimension(1:N_P) :: vel_x,vel_y,vel_z
real(rp),dimension(1:N_P) :: fvel_x,fvel_y,fvel_z
real(rp),dimension(1:N_P) :: rad
!
integer :: p
!
integer :: it,it_min,it_max
real(rp) :: t,dt,t_p
!
real(rp) :: y_p,u_p,s_p
real(rp) :: T_lp,T_l,T_e
real(rp) :: Stk
real(rp) :: vel_r_x,vel_r_y,vel_r_z,vel_r
real(rp) :: Re_p,tau_p,f_p,tau_l
real(rp) :: dsdy,duudy
real(rp) :: dcsi_1,dcsi_2,dcsi_3
real(rp) :: sigma_x,sigma_y,sigma_z
real(rp) :: rhs_pos_x,rhs_pos_y,rhs_pos_z
real(rp) :: rhs_vel_x,rhs_vel_y,rhs_vel_z
real(rp) :: rhs_fvel_x,rhs_fvel_y,rhs_fvel_z
!
character(32) :: fileres
character(256) :: open_file
!
character(32),parameter :: open_path="./data/"
character(32),parameter :: fileres_01="prt_data_1234567.bin"
!
!======================================================================
!Initial Conditions
!
call srand(1)
!
do p=1,N_P
  !
  rad(p)=radius
  !
  pos_x(p)=(L_x-1.0d-10)*rand()+0.5d-10
  pos_y(p)=(L_y-1.0d-10)*rand()+0.5d-10
  pos_z(p)=(L_z-1.0d-10)*rand()+0.5d-10
  !
  vel_x(p)=0.0_rp
  vel_y(p)=0.0_rp
  vel_z(p)=0.0_rp
  !
  fvel_x(p)=0.0_rp
  fvel_y(p)=0.0_rp
  fvel_z(p)=0.0_rp
  !
enddo
!
it_min=0
!
it_max=7000000
!
dt=1.0d-5
!
t=0.0_rp
!
fileres=fileres_01
!
call PrintStep(it_min)
!
call PrintPath
!
call SaveData
!
!Main Block
do it=it_min+1,it_max
  !
  t=t+dt
  t_p=t*u_tau**2.0_rp/nu
  !
  if(mod(it,schermo)==0) then
    !
    print*,"It = ",it," t = ",t,t_p
    !
  endif
  !
  !Adimensional Coordinate Definition
  do p=1,N_P
    !
    if(pos_y(p)>L_y/2.0_rp) then
  !
  y_p=abs((L_y-pos_y(p))/l_tau)
  !
  s_p=-1.0_rp
  !
else
  !
  y_p=abs(pos_y(p)/l_tau)
  !
  s_p=+1.0_rp
  !
endif
!
u_p=1.0_rp/vk_c*log(1.0_rp+vk_c*y_p)+C_k*(1.0_rp-exp(-y_p/11.0_rp)\\
-y_p/11.0_rp*exp(-0.33_rp*y_p))
!
!Particle Statistics Computation
call CmptSigmaX(y_p,sigma_x)
call CmptSigmaY(y_p,sigma_y)
call CmptSigmaZ(y_p,sigma_z)
!
call CmptdSigmadY(y_p,dsdy)
!
call CmptdUUdY(y_p,duudy)
!
call GenGaussian(sqrt(dt),0.0_rp,dcsi_1)
call GenGaussian(sqrt(dt),0.0_rp,dcsi_2)
call GenGaussian(sqrt(dt),0.0_rp,dcsi_3)
!
if(y_p<5.0_rp) then
  !
  T_lp=10.0_rp
  !
elseif((y_p>=5.0_rp).and.(y_p<200.0_rp)) then
  !
  T_lp=7.122_rp+0.5731_rp*y_p-0.00129_rp*y_p**2
  !
else
  !
  T_lp=70.142_rp
  !
endif
!
T_l=T_lp*nu/u_tau**2
T_e=T_l/beta
!
tau_p=1.0_rp/18.0_rp*rho_r*(2.0_rp*rad(p))**2/nu
tau_l=T_l/beta*(1.0_rp-(1.0_rp-beta)*(1.0_rp+tau_p/T_e)\\
**(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_e)))
!
Stk=tau_p/tau_l
!
!Fluctuation Component Computation
rhs_fvel_x=sqrt(2.0_rp/tau_l)*dcsi_1+duudy*dt/(1.0_rp+Stk)
```

```
      rhs_fvel_y=sqrt(2.0_rp/tau_l)*dcsi_2+s_p*dsdy*dt/(1.0_rp+Stk)
      rhs_fvel_z=sqrt(2.0_rp/tau_l)*dcsi_3
      !
      if(y_p<=150.0_rp) then
        !
        vel_r_x=u_p*u_tau+fvel_x(p)*sigma_x-vel_x(p)
        vel_r_y=          fvel_y(p)*sigma_y-vel_y(p)
        vel_r_z=          fvel_z(p)*sigma_z-vel_z(p)
        !
      else
        !
        vel_r_x=u_p*u_tau-vel_x(p)
        vel_r_y=         -vel_y(p)
        vel_r_z=         -vel_z(p)
        !
      endif
      !
      !Particle Motion Computation
      vel_r=sqrt(vel_r_x**2+vel_r_y**2+vel_r_z**2)
      !
      tau_p=2.0_rp/9.0_rp*rho_r*rad(p)**2/nu
      !
      Re_p=2.0_rp*vel_r*rad(p)/nu
      !
      f_p=1.0_rp+0.15_rp*Re_p**0.687_rp
      !
      rhs_vel_x=f_p*vel_r_x/tau_p
      rhs_vel_y=f_p*vel_r_y/tau_p
      rhs_vel_z=f_p*vel_r_z/tau_p
      !
      rhs_pos_x=vel_x(p)
      rhs_pos_y=vel_y(p)
      rhs_pos_z=vel_z(p)
      !
      pos_x(p)=pos_x(p)+vel_x(p)*dt
      pos_y(p)=pos_y(p)+vel_y(p)*dt
      pos_z(p)=pos_z(p)+vel_z(p)*dt
      !
      vel_x(p)=vel_x(p)+rhs_vel_x*dt
      vel_y(p)=vel_y(p)+rhs_vel_y*dt
      vel_z(p)=vel_z(p)+rhs_vel_z*dt
      !
      if(y_p<=150.0_rp) then
        !
        fvel_x(p)=(fvel_x(p)+rhs_fvel_x)/(1.0_rp+dt/tau_l)
        fvel_y(p)=(fvel_y(p)+rhs_fvel_y)/(1.0_rp+dt/tau_l)
        fvel_z(p)=(fvel_z(p)+rhs_fvel_z)/(1.0_rp+dt/tau_l)
        !
      else
        !
        fvel_x(p)=0.0_rp
        fvel_y(p)=0.0_rp
        fvel_z(p)=0.0_rp
        !
```

```fortran
    endif
    !
  enddo
  !
  !End Cycle
  call ReposPart
  !
  if(mod(it,stampa)==0) then
    !
    fileres=fileres_01
    !
    call PrintStep(it)
    !
    call PrintPath
    !
    call SaveData
    !
  endif
  !
enddo
!
contains
!
!======================================================================
!
subroutine SaveData
  implicit none
  integer :: fid
  integer :: p
  !
  open(newunit=fid,file=open_file,status="replace")
  !
  write(fid,*) N_P
  !
  do p=1,N_P
    !
    write(fid,*) pos_x(p),pos_y(p),pos_z(p),vel_x(p),vel_y(p),vel_z(p)
    !
  enddo
  !
  close(fid)
  !
  return
end subroutine SaveData
!
!======================================================================
!
subroutine ReposPart
  implicit none
  integer :: p
  !
  do p=1,N_P
    !
    if(pos_x(p)<0.0_rp) pos_x(p)=pos_x(p)+L_x
    if(pos_x(p)>L_x   ) pos_x(p)=pos_x(p)-L_x
    !
    if(pos_z(p)<0.0_rp) pos_z(p)=pos_z(p)+L_z
    if(pos_z(p)>L_z   ) pos_z(p)=pos_z(p)-L_z
    !
    if(pos_y(p)<rad(p)) then
      !
      pos_y(p)=2.0_rp*rad(p)-pos_y(p)
      vel_y(p)=-vel_y(p)
      !
    endif
    !
    if(pos_y(p)>L_y-rad(p)) then
      !
      pos_y(p)=2.0_rp*L_y-2.0_rp*rad(p)-pos_y(p)
      vel_y(p)=-vel_y(p)
      !
    endif
    !
  enddo
  !
  return
end subroutine ReposPart
!
!=====================================================================
!
subroutine GenGaussian(var,mean,sigma)
  implicit none
  real(rp),intent(in) :: var,mean
  real(rp),intent(out) :: sigma
  !
  real(rp) :: uni_1,uni_2
  real(rp) :: rnd_1,rnd_2
  !
  call random_number(rnd_1)
  call random_number(rnd_2)
  !
  uni_1=(1.0_rp-1.0d-14)*rnd_1+1.0d-14
  uni_2=(1.0_rp-1.0d-14)*rnd_2+1.0d-14
  !
  sigma=var*sqrt(-2.0_rp*log(uni_1))*cos(2.0_rp*pi*uni_2)+mean
  !
  if(sigma>mean+5.0_rp*var) sigma=mean+5.0_rp*var
  if(sigma<mean-5.0_rp*var) sigma=mean-5.0_rp*var
  !
  return
end subroutine GenGaussian
!
!=====================================================================
!
subroutine CmptSigmaX(y_plus,sigma)
  implicit none
  real(rp),intent(in) :: y_plus
  real(rp),intent(out) :: sigma
  !
  integer :: i_c,i_m,i_p
  !
  real(rp) :: m,q
  !
  real(rp),dimension(1:152),parameter :: x=[&
  ........]
  !
  real(rp),dimension(1:152),parameter :: y=[&
 ........]
  !
  if(y_plus<=150.0_rp) then
    !
    i_c=minloc(abs(y_plus-x),1)
    !
    if(x(i_c)<y_plus) then
      !
      i_m=i_c
      i_p=i_c+1
      !
    else
      !
      i_m=i_c-1
      i_p=i_c
      !
    endif
    !
    m=(y(i_p)-y(i_m))/(x(i_p)-x(i_m))
    q=y(i_m)-m*x(i_m)
    !
    sigma=(m*y_plus+q)*u_tau
    !
  else
    !
    sigma=0.0_rp
    !
  endif
  !
  return
end subroutine CmptSigmaX
!
!=======================================================================
!
subroutine PrintStep(iostep)
  implicit none
  integer,intent(in) :: iostep
  !
  write(fileres(10:16),1) iostep
1 format(1I7.7)
  !
  return
end subroutine PrintStep
!
! ========================================================================
  !
  subroutine PrintPath
    implicit none
    !
    open_file=trim(open_path)//trim(fileres)
    !
    return
  end subroutine PrintPath
  !
  ! =======================================================================
  !
end program ParticleWallModel
```

7.1.2 Re_τ=550 Modifications

```
 !Particle Statistics Computation
      call CmptSigmaX(y_p,sigma_x)
      call CmptSigmaY(y_p,sigma_y)
      call CmptSigmaZ(y_p,sigma_z)
      !
      call CmptdSigmadY(y_p,dsdy)
      !
      call CmptdUUdY(y_p,duudy)
      !
      call GenGaussian(sqrt(dt),0.0_rp,dcsi_1)
      call GenGaussian(sqrt(dt),0.0_rp,dcsi_2)
      call GenGaussian(sqrt(dt),0.0_rp,dcsi_3)
      !
      !
      if(y_p<10.0_rp) then
        !
        T_lpx=10.0_rp
        T_lpy=11.0_rp
        T_lpz=14.0_rp
        !
      elseif((y_p>=10.0_rp).and.(y_p<550.0_rp)) then
        !
        call CmptTaux(y_p,T_lpx)
        call CmptTauy(y_p,T_lpy)
        call CmptTauz(y_p,T_lpz)
        !
      endif
      !
      T_lx=T_lpx*nu/u_tau**2
      T_ly=T_lpy*nu/u_tau**2
      T_lz=T_lpz*nu/u_tau**2
      T_ex=T_lx/beta
      T_ey=T_ly/beta
      T_ez=T_lz/beta
      !
```

```
        tau_p=1.0_rp/18.0_rp*rho_r*(2.0_rp*rad(p))**2/nu
        tau_lx=T_lx/beta*(1.0_rp-(1.0_rp-beta)*(1.0_rp+tau_p/T_ex)\\
        **(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ex)))
        tau_ly=T_ly/beta*(1.0_rp-(1.0_rp-beta)*(1.0_rp+tau_p/T_ey)\\
        **(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ey)))
        tau_lz=T_lz/beta*(1.0_rp-(1.0_rp-beta)*(1.0_rp+tau_p/T_ez)\\
        **(-0.4_rp*(1.0_rp+0.01_rp*tau_p/T_ez)))
        !
        Stk=tau_p/((tau_lx+tau_ly+tau_lz)/3.0_rp)
```

References

[1] B. A. Lectures on applied computational fluid dynamics. [Online]. Available: https://www.bakker.org/cfm/publications/Lectures-Applied-CFD.pdf

[2] J. D. Anderson, *Fundamentals of aerodynamics*, 5th ed. McGraw-Hill, Feb. 2011. [Online]. Available: http://www.worldcat.org/isbn/9780073398105

[3] B. Arcen, A. Tanière, and L. Zaichik, "Assessment of a statistical model for the transport of discrete particles in a turbulent channel flow," *International Journal of Multiphase Flow - INT J MULTIPHASE FLOW*, vol. 34, pp. 419–426, 04 2008.

[4] M. Bernardini, "Reynolds number scaling of inertial particle statistics in turbulent channel flows," *Journal of Fluid Mechanics*, vol. 758, 11 2014.

[5] M. Bernardini, S. Pirozzoli, and P. Orlandi, "Velocity statistics in turbulent channel flow up to reτ =4000," *Journal of Fluid Mechanics*, vol. 742, 02 2014.

[6] S. T. Bose and G. I. Park, "Wall-modeled large-eddy simulation for complex turbulent flows," *Annual Review of Fluid Mechanics*, vol. 50, no. 1, pp. 535–561, 2018.

[7] A. Dehbi, "Turbulent particle dispersion in arbitrary wall-bounded geometries: A coupled cfd-langevin-equation based approach," *International Journal of Multiphase Flow*, vol. 34, no. 9, pp. 819–828, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301932208000463

[8] ——, "Validation against dns statistics of the normalized langevin model for particle transport in turbulent channel flows," *Powder Technology*, vol. 200, no. 1, pp. 60–68, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0032591010000835

[9] N. El Gharbi, R. Absi, A. Benzaoui, and R. Bennacer, “An improved near-wall treatment for turbulent channel flows,” *International Journal of Computational Fluid Dynamics*, vol. 25, pp. 41–46, 01 2011.

[10] I. Iliopoulos, Y. Mito, and T. Hanratty, “A stochastic model for solid particle dispersion in a nonhomogeneous turbulent field,” *International Journal of Multiphase Flow*, vol. 29, pp. 375–394, 03 2003.

[11] A. S. Jebakumar, K. N. Premnath, V. Magi, and J. Abraham, “Fully-resolved direct numerical simulations of particle motion in a turbulent channel flow with the lattice-boltzmann method,” *Computers Fluids*, vol. 179, pp. 238–247, 2019.

[12] P. Johnson, M. Bassenne, and P. Moin, “Turbophoresis of small inertial particles: theoretical considerations and application to wall-modelled large-eddy simulations,” *Journal of Fluid Mechanics*, vol. 883, 01 2020.

[13] G. Kallio and M. Reeks, “A numerical simulation of particle deposition in turbulent boundary layers,” *International Journal of Multiphase Flow*, vol. 15, no. 3, pp. 433–446, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0301932289900128

[14] P. Kundu, I. Cohen, and D. Dowling, *Fluid Mechanics*, ser. Academic Press. Academic Press, 2015. [Online]. Available: https://books.google.it/books?id=uYevoQEACAAJ

[15] Y. Liu, *CFD SIMULATION OF PHASE PARTICLE ENTRAPMENT Chapter*, 01 2011.

[16] C. Marchioli, A. Soldati, J. Kuerten, B. Arcen, A. Tanière, G. Goldensoph, K. Squires, M. Cargnelutti, and L. Portela, “Statistics of particle dispersion in direct numerical simulations of wall-bounded turbulence: Results of an international collaborative benchmark test,” *International Journal of Multiphase Flow*, vol. 34, no. 9, pp. 879–

893, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301932208000414

[17] F. Nicoud, F. Ducros, "Subgrid-scale stress modelling based on the square of the velocity gradient tensor," *Flow, Turbulence and Combustion*, vol. 62, p. 183–200, 1999.

[18] D. Perrone, "A lagrangian view of turbulent mixing in channel flow through complex networks," 2019.

[19] D. A. Philips, R. Rossi, and G. Iaccarino, "Large-eddy simulation of passive scalar dispersion in an urban-like canopy," *Journal of Fluid Mechanics*, vol. 723, p. 404–428, 2013.

[20] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2000.

[21] L. F. Richardson and P. Lynch, *Weather Prediction by Numerical Process*, 2nd ed., ser. Cambridge Mathematical Library. Cambridge University Press, 2007.

[22] H. C. Rodean, "Notes on the langevin model for turbulent diffusion of "marked" particles," University of North Texas Librariess, Tech. Rep., 1994.

[23] S. Rodriguez, *LES and DNS Turbulence Modeling*. Cham: Springer International Publishing, 2019, pp. 197–223.

[24] M. S. Shadloo, D. Le Touzé, and G. Oger, "Smoothed particle hydrodynamics method for fluid flows, towards industrial applications-motivations, current state, and challenges," 06 2016.

[25] S. Subramaniam and S. Balachandar, "1 - introduction," in *Modeling Approaches and Computational Methods for Particle-Laden Turbulent Flows*, ser. Computation and Analysis of Turbulent Flows, S. Subramaniam and S. Balachandar, Eds. Academic Press, 2023, pp. 1–42.

[26] A. Y. Varaksin, Ed., *Concise Information About Single-Phase and Heterogeneous Turbulent Flows*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–25. [Online]. Available: https://doi.org/10.1007/978-3-540-68054-3_1

Acknowledgments

Per concludere non posso non dedicare due parole a tutte le persone che in un modo o nell'altro mi hanno aiutato e mi sono state affianco durante questo percorso e negli ultimi anni.
Prima di tutto ringrazio il mio relatore Prof. Francesco Picano per la sua disponibilità, e il mio correlatore Prof. Federico Dalla Barba per l'enorme aiuto con lo sviluppo del benedetto simulatore, nonostante si perdessero i ”contatti” ogni tanto.

Partiamo ora con i pezzi grossi: grazie a mamma e papà, non so quanto siate riusciti a stare dietro alle mie vicissitudini nel laurearmi, ma nonostante ciò mi avete comunque sempre appoggiato a prescindere, grazie; grazie ai nonni, di cui sento sempre il supporto nonostante i 600 e passa km di distanza, ora che è finita questa storia prometto di chiamarvi più spesso; grazie anche a mia sorella Alessia, insperatamente ti sei rivelata una buona revisionatrice di manoscritti e anche utile per fare grafici, senza i momenti ”leggeri” a tavola, probabilmente sarei impazzito due mesi fa; e un grazie obbligatorio anche a Romeo, basta che la smette di rompere vasi.

C'è poi un gruppo di persone che avrebbe bisogno di una sezione dedicata nell'indice: i miei amici. Mai come durante la stesura dei ringraziamenti mi sono reso conto di quante persone ho avuto modo di e posso considerare amiche.
Grazie a Fabio, Gian e Carlo, chissà cosa sarebbe successo se non fossimo arrivati giusti giusti il primo giorno di lezione ben 6 anni fa. Sicuramente avrei fatto meno colazioni con brioche e lavorato molto di più nel progettino di Lorenzini, meglio non pensarci troppo.
Grazie anche a Sam, Carmen, Francesca e tutti i miei altri amichetti universitari, senza i quali sarebbe stato sicuramente tutto più noioso.

Per gli amici del Dolo, Pier, Giovi, Novi e tutti gli altri, grazie per tutte le serate e le vacanze passate assieme, siamo davvero un signor gruppo.
Gruppo capitanato da Federica, che non posso non ringraziare, perchè dopo 5 anni alle superiori dove mi ha snobbato, a 20 anni ha deciso che potevamo essere finalmente amici, per fortuna.
Infine grazie a Giulio: nonostante i nostri percorsi siano stati completamente diversi, abbiamo sempre condiviso tantissimo. Grazie alle nostre chiacchierate e passeggiate sono rimasto sano di mente in questi anni, anzi credo di essere costantemente cresciuto come persona e spero di aver ”causato” questo anche in te.
Menzione d'onore a Badr, che pure da Parigi ha cercato di aiutarmi con Python e alleggerire le serate giocando assieme. Grazie davvero, speriamo di poterci rivedere presto.

E per concludere il grazie più grande che meriterebbe un capitolo dedicato.
A te che fino a non molto tempo fa pensavi di essere dentro il gruppo di amici sopracitato, a mia insaputa.
Giorgia, un paio di anni fa sei entrata nella mia vita a cannone, chiaramente intenzionata a riprenderti tutti quegli anni dove non ti ho calcolato mai. Mi hai raccolto in un momento veramente nero per me, me ne hai tirato fuori, ed è grazie soprattutto al tuo sostegno se poi sono riuscito a finire questo percorso. Questa è una cosa importantissima e indelebile, sono felicissimo che possiamo dire di esserci laureati assieme.
Non ho le parole nè le capacità intellettuali per esprimere quanto sono grato di tutto ciò, o quanto sia felice che tu sia piombata nella mia testa e nel mio cuore.
Spero di poterti ripagare con la stessa moneta anche ora che siamo finalmente fuori e liberi di fare quello che vogliamo.

Grazie di cuore a te.

Grazie di cuore a tutti.