Doctoral Thesis

Study of the flow field through the wall of a Diesel particulate filter using Lattice Boltzmann Methods.

A dissertation presented
by

José Pedro García Galache

Supervised
by

Prof. Dr. Ing. Antonio Gil Megías

at

Departamento de Máquinas y Motores Térmicos

Submitted
to

Universitat Politècnica de València

in partial fulfilment of the requirements for the degree of Doctor of Philosophy

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

September 2017

Resumen

La contaminación se está volviendo un gran problema para las grandes áreas metropolitanas, en gran parte debido al tráfico. A nivel europeo, al igual que en otras áreas, la regulación es cada vez más restrictiva. Una buena prueba de ello es la normativa Euro de la Unión Europea.

Especialmente importantes son las emisiones de óxidos de nitrógeno (NOx) y partículas (PM). La reducción de contaminantes se puede abordar desde dos estrategias distintas. La primera es la prevención. Modificar el proceso de combustión a través de las leyes de inyección o controlar la renovación de la carda son los métodos más comunes. La segunda estrategia es la eliminación. Se puede reducir los NOx mediante catálisis o atmósfera reductora y las partículas mediante la instalación de un filtro en el conducto de escape. La presente tesis se centra en el estudio de éste último.

La mayoría de as estrategias para la reducción de emisiones penalizan el consumo. El filtro de partículas no es una excepción. Restringe el paso de aire. Como consecuencia la presión se incrementa a lo largo de toda la línea reduciendo las prestaciones del motor. La optimización del filtro es de vital importancia. Tiene que mantener su eficacia a la par que que se minimiza la caída de presión y con ella el consumo de combustible. El objetivo de la tesis es encontrar la relación entre la miscroestructura y las propiedades macroscópicas del filtro. Las conclusiones del estudio podrán utilizarse para optimizar la microestructura.

La microestructura elegida imita los filtros de mulita acicular. Se genera por ordenador mediante generación procedimental utilizando parámetros aleatorios. Gracias a ello se puede estudiar la relación que existe entre la microestructura y las propiedades macroscópicas como la porosidad y la permeabilidad. El campo fluido se resuelve con LaBMoTer, un software desarrollado en esta tesis. Está basado en Lattice Boltzmann, una nueva aproximación para simular fluidos. Además también se ha utilizado el framework waLBerla desarrollado por la universidad Friedrich Alexander de Erlangen Nürnberg.

La segunda parte de la tesis se centra en las partículas suspendidas en el fluido. Sus propiedades vienen dadas en función del diámetro aerodinámico. Es una buena aproximación desde un punto de vista macroscópico. Sin embargo éste no es el caso. El tamaño de la discretización requerida para calcular el medio poroso es similar al tamaño de las partículas. En consecuencia se necesita simular geometrías realistas. Las partículas Diesel son agregados de esferas. El proceso de aglomeración se ha simulado mediante colisión balística. Los resultados se han analizado con detalle.

El segundo paso es la caracterización aerodinámica de los aglomerados.

Debido a que el tamaño de las partículas precursoras es similar a la distancia entre moléculas el fluido no puede ser considerado un medio continuo. Se necesita una nueva aproximación. La herramienta apropiada es la Simulación Directa Monte Carlo (DSMC). Por ello se ha desarrollado un software basado en esta formulación. Desafortunadamente no ha habido tiempo suficiente como para implementar condiciones de contorno sobre geometrías complejas.

La tesis ha sido fructífera en múltiples aspectos. Se ha desarrollado un modelo basado en generación procedimental capaz de crear una microestructura que aproxime mulita acicular. Se ha implementado y validado un nuevo solver CFD, LaBMoTer. Además se ha planteado una técnica que optimiza la preparación del cálculo. El proceso de aglomeración se ha estudiado en detalle gracias a un nuevo simulador desarrollado ad hoc para esta tarea. Mediante el análisis estadístico de los resultados se han planteado modelos que reproducen la población de partículas y su evolución con el tiempo. Técnicas de Cuantificación de Incertidumbre se han empleado para modelar la dispersión de datos. Por último, un simulador basado en DSMC se ha desarrollado para calcular fluidos rarificados.

Resum

La contaminació s'està tornant un gran problema per a les grans àrees metropolitanes, en gran part degut al tràfic. A nivell europeu, a l'igual que en atres àrees, la regulació és cada volta més restrictiva. Una bona prova d'això és la normativa Euro de l'Unió Europea.

Especialment importants són les emissions d'òxits de nitrogen (NOx) i partícules (PM). La reducció de contaminants se pot abordar des de dos estratègies distintes. La primera és la prevenció. Modificar el procés de combustió a través de les lleis d'inyecció o controlar la renovació de la càrrega són els mètodos més comuns. La segona estratègia és l'eliminació. Se pot reduir els NOx mediant catàlisis o atmòsfera reductora i les partícules mediant l'instalació d'un filtre en el vas d'escap. La present tesis se centra en l'estudi d'este últim.

La majoria de les estratègies per a la reducció d'emissions penalisen el consum. El filtre de partícules no és una excepció. Restringix el pas d'aire. Com a conseqüència la pressió s'incrementa a lo llarc de tota la llínea reduint les prestacions del motor. L'optimisació del filtre és de vital importància. Ha de mantindre la seua eficàcia a la par que que es minimisa la caiguda de pressió i en ella el consum de combustible. L'objectiu de la tesis és trobar la relació entre la microescritura i les propietats macroscòpiques del filtre. Les conclusions de l'estudi podran utilisar-se per a optimisar la microestructura.

La microestructura elegida imita els filtres de mulita acicular. Se genera per ordenador mediant generació procedimental utilisant paràmetros aleatoris. Gràcies ad això es pot estudiar la relació que existix entre la microestructura i les propietats macroscòpiques com la porositat i la permeabilitat. El camp fluït se resol en LaBMoTer, un software desenrollat en esta tesis. Està basat en Lattice Boltzmann, una nova aproximació per a simular fluïts. Ademés també s'ha utilisat el framework waLBerla, desentollat per l'Universitat Friedrich Alexander d'Erlangen Nürnberg.

La segona part de la tesis se centra en les partícules suspeses en el fluït. Les seues propietats venen donades en funció del diàmetro aerodinàmic. És una bona aproximació des d'un punt de vista macroscòpic. No obstant este no és el cas. El tamany de la discretisació requerida per a calcular el mig porós és similar al tamany de les partícules. En conseqüència es necessita simular geometries realistes. Les partícules diésel són agregats d'esferes. El procés d'aglomeració s'ha simulat mediant colisió balística. Els resultats s'han analisat en detall.

El segon pas és la caracterisació aerodinàmica dels aglomerats. Degut a que el tamany de les partícules precursores és similar a la distància entre molècules

el fluït no pot ser considerat un mig continu. Se necessita una nova aproximació. La ferramenta apropiada és la Simulació Directa Monte Carlo (DSMC). Per això s'ha desenrollat un software basat en esta formulació. Malafortunadament no ha hagut temps suficient com per a implementar condicions de contorn sobre geometries complexes.

La tesis ha segut fructífera en múltiples aspectes. S'ha desenrollat un model basat en generació procedimental capaç de crear una microestructura que aproxime mulita acicular. S'ha implementat i validat un nou solver CFD, LABMOTER. Ademés s'ha plantejat una tècnica que optimisa la preparació del càlcul. El procés d'aglomeració s'ha estudiat en detall gràcies a un nou simulador desenrollat ad hoc per ad esta tasca. Mediant l'anàlisis estadístic dels resultats s'han plantejat models que reproduixen la població de partícules i la seua evolució en el temps. Tècniques de Quantificació d'Incertea s'han empleat per a modelar la dispersió de senyes. Per últim, un simulador basat en DSMC s'ha desenrollat per a calcular fluïts rarificats.

Abstract

Contamination is becoming an important problem in great metropolitan areas. A large portion of the contaminants is emitted by the vehicle fleet. At European level, as well as in other economical areas, the regulation is becoming more and more restrictive. Euro regulations are the best example of this tendency.

Specially important are the emissions of nitrogen oxide (NOx) and Particle Matter (PM). Two different strategies exist to reduce the emission of pollutants. One of them is trying to avoid their creation. Modifying the combustion process by means of different fuel injection laws or controlling the air regeneration are the typical methods. The second set of strategies is focused on the contaminant elimination. The NOx are reduced by means of catalysis and/or reducing atmosphere, usually created by injection of urea. The particle matter is eliminated using filters. This thesis is focused in this matter.

Most of the strategies to reduce the emission of contaminants penalise fuel consumption. The particle filter is not an exception. Its installation, located in the exhaust duct, restricts the pass of the air. It increases the pressure along the whole exhaust line before the filter reducing the performance. Optimising the filter is then an important task. The efficiency of the filter has to be good enough to obey the contaminant normative. At the same time the pressure drop has to be as low as possible to optimise fuel consumption and performance. The objective of the thesis is to find the relation between the micro-structure and the macroscopic properties. With this knowledge the optimisation of the micro-structure is possible.

The micro-structure of the filter mimics acicular mullite. It is created by procedural generation using random parameters. The relation between microstructure and the macroscopic properties such as porosity and permeability are studied in detail. The flow field is solved using LaBMoTer, a software developed during this thesis. The formulation is based on Lattice Botlzmann Methods, a new approach to simulate fluid dynamics. In addition, waLBerla framework is used to solve the flow field too. This tool has been developed by Friedrich Alexander University of Erlangen Nürnberg.

The second part of the thesis is focused on the particles immersed into the fluid. The properties of the particles are given as a function of the aerodynamic diameter. This is enough for macroscopic approximations. However, the discretisation of the porous media has the same order of magnitude than the particle size. Consequently realistic geometry is necessary. Diesel particles are aggregates of spheres. A simulation tool is developed to create these aggregated using ballistic collision. The results are analysed in detail.

The second step is to characterise their aerodynamic properties. Due to the small size of the particles, with the same order of magnitude than the separation between molecules of air, the fluid can not be approximated as a continuous medium. A new approach is needed. Direct Simulation Monte Carlo is the appropriate tool. A solver based on this formulation is developed. Unfortunately complex geometries could not be implemented on time.

The thesis has been fruitful in several aspects. A new model based on procedural generation has been developed to create a micro-structure which mimics acicular mullite. A new CFD solver based on Lattice Boltzmann Methods, LaBMoTer, has been implemented and validated. At the same time it is proposed a technique to optimised setup. Ballistic agglomeration process is studied in detail thanks to a new simulator developed ad hoc for this task. The results are studied in detail to find correlation between properties and the evolution in time. Uncertainty Quantification is used to include the Uncertainty in the models. A new Direct Simulation Monte Carlo solver has been developed and validated to calculate rarefied flow.

Agradecimientos

Todo en la vida de una persona, desde el más mínimo detalle hasta el más gran acontecimiento, influye en el desarrollo personal y define quién se es hoy en día. Es por ello por lo que he de agradecer a todo (circunstancias) y todos (personas) quién soy y, por extensión, la escritura de esta tesis. Sin embargo hay un grupo de gente que me ha ayudado más de cerca y que por ello he de mencionarlos en estas líneas de forma expresa.

En primer lugar está el CMT-Motores Térmicos como institución y la gente que lo forma. No hace falta decir que sin los recursos materiales y humanos puestos a mi disposición no podría haber realizado esta tesis con éxito. Por eso he de agradecer a Franciso Payri y a José María Desántes la oportunidad de haber realizado la tesis en el departamento. Ese edificio ha sido mi segunda casa (yo diría que más bien la primera) durante más de cuatro años y la gente que en él trabaja mi *"familia académica"*. Por supuesto la persona que más ha influido es mi tutor, el profesor Antonio Gil. Él es quien me ha guiado en tan ardua tarea. He de agradecer la confianza que en mí ha depositado, pues al tomar decisiones siempre ha tenido en cuenta mis opiniones, valoraciones y puntos de vista. Esta tesis ha sido una apuesta por la innovación que podría haberse perdido y Antonio apostó por mí.

Además de mi tutor también me han tenido que soportar otros compañeros que ahora también considero amigos. Pau Raga es el campeón en esta materia. Él ha sido testigo de toda mi etapa de doctorando. Me enseño el funcionamiento interno del CMT-Motores Térmicos y por ello he de estarle agradecido. Por supuesto también he de acordarme de mis otros compañeros. Pedro Quintero y Sebastián han sufrido el CFD tanto como yo (o más). Eso une mucho. Uno espero que sea un gran profesor y el otro ha de estar acabando la tesis para ser un gran profesional. Otras personas que en mayor o menor grado también me han enseñado algo son los numerosos proyectandos que han completado su formación en la oficina. Desde el primer año, cuando eran muchos y competían entre ellos por la mejor silla, hasta el último. Sé que me olvidaré de alguno porque la lista es larga pero que menos que mencionarlos

para darles las gracias de forma explícita. Gracias Carlos, Luis, Javi, Pablo, Juanjo, Nacho, David, Carmen, Raúl, Sergio, Ainhoa, Enric, Salva, Miguel, Adrián, Juan Carlos, Iván y algún otro que por seguro se me olvida y ruego que me disculpe.

Además de mis compañeros de oficina hay otros miembros de CMT-Motores Térmicos a los que les he de agradecer los buenos ratos pasados y con los que he compartido vivencias. A los primeros que conocí fue a mis compañeros de carrera Marcos, Juanra y Pedro. Al entrar en CMT el grupo creció con Ricardo, los tres Danis y Ausias entre otros. Hay otras personas que también me han ayudado de forma paralela como el personal de secretaría, siempre pendiente del papeleo que debíamos hacer, y el profesorado, gracias al cuál he podido completar mi formación como ingeniero.

Durante mi etapa de doctorando he vivido en Erlangen durante un periodo de tres meses. Una experiencia enriquecedora y productiva en todos los aspectos. Tengo que agradecer al profesor Ulrich Rüde la oportunidad que me ha dado de trabajar con su equipo. Un grupo de gente magnifica, agradable y grandes profesionales. Muchos de ellos me ayudaron tanto en el terreno profesional como en el personal. Fruto del trabajo en equipo con Christian Godenschwager tengo una gran publicación. También tengo que agradecer la amabilidad de Christian Kuschel y Dominik, con quienes pasé muy buenos momentos. Otros compañeros de los que tengo muy buenos recuerdos son Christoph, Florian, Sebastian, Kristina, Regina, Ehsan, Julian... Por supuesto también he de acordarme de quienes me ayudaron con las gestiones necesarias allí, Iris y Frank. No puedo dejar fuera de los agradecimientos al Welcome Center. Un organismo perteneciente a la universidad que ayuda al personal desplazado a instalarse en la ciudad. Gracias a ellos tuve la oportunidad de visitar Rothemburg ob der Tauber y de paso disfrutar de la magnifica compañía de Lucija Kozeljnik. Ella es la persona que me recibió y que primero conocí en Alemania. Llena de empatía y afabilidad, desempeña de forma rigurosa su cometido. Gracias Lucija por tu magnifica atención y ayuda y por tu conversación agradable.

De entre todos los profesores que me han impartido clase durante la carrera hay dos que me han marcado especialmente y que han hecho que reoriente mi carrera profesional. Uno de ellos es Guillermo Paniagua. Excelente profesor y aun mejor persona que sin duda potencia la creatividad y la capacidad de innovación manteniendo siempre motivado al alumnado. Gracias a él tuve la oportunidad de estudiar en el instituto von Kàrmàn y así, a la par de recibir una excelente formación, pude conocer a un grupo excepcional de personas. El otro es el gran experto en supercomputación Sergio Hoyas. Una asignatura optativa fue suficiente como para hacer aflorar todo un potencial que tenía escondido. Gracias maestro por haberme enseñado las bases de conocimiento necesarias para desarrollar la presente tesis doctoral y lo que es mi actual oficio.

Por supuesto hay muchas más personas que me han ayudado en múltiples aspectos y en situaciones diversas. Como ya dije al comienzo de este escrito todo quien me ha rodeado ha influido en mayor o menor grado en quién soy hoy. Soy consciente de este hecho y de que el no nombrar a todos es en cierto modo una injusticia. Gracias familiares, amigos, profesores... gracias a todos aquellos que me habéis convertido en quien soy hoy.

Pero de entre todas las personas que me han ayudado a quien más tengo que agradecer es a mi familia. Sin ellos no sería ni la sombra de quien soy ahora. Gracias Consuelo y Felicísimo por ser mi guía en este mundo, mi ejemplo a seguir. Gracias por haberme guiado comprendiendo y respetando siempre mis valores y mi particular forma de pensar, y por enseñarme de igual forma a comprender y respetar las particularidades y valores de los demás. Gracias por el trato recibido, por saciar mi curiosidad durante mi infancia y potenciar las capacidades que hoy en día me caracterizan. En resumen, gracias por mi educación y el apoyo incondicional recibidos. También tengo que agradecer el trato recibido por mi tío Adolfo, mi segundo padre. Gracias por haber desempeñado funciones que en un principio no están encomendadas para un tío pero que amablemente sí has realizado. De igual manera tengo que dar las gracias a mi hermano Adolfo Andrés. Gracias hermano por tu apoyo y confianza, por ayudarme cuando lo necesito y por supuesto, por haberme introducido al fascinante mundo de Linux.

Official Acknowledgements

J.P.G. Galache was supported by a research grant from Generalitat Valenciana (Programa VALI+d, ACIF/2014/020 and BEFPI/2015/006).

This work has been supported by the PAID-06-14 program (Primeros Proyectos de Investigación UPV) of Universitat Politècnica de València. “ESTUDIO DEL FLUJO EN MEDIOS POROSOS MEDIANTE MÉTODOS LATTICE-BOLTZMANN”

The author gratefully acknowledges the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ, www.lrz.de).

Contents

1 Introduction **1**

1.1 Motivation and objectives 1

1.2 Justification and previous work 2

1.3 Methodology 6

1.3.1 Work philosophy 6

1.3.2 Work flow 7

1.4 Resources and tools 9

2 The porous media and its nature **15**

2.1 Description 15

2.2 The filtering process 16

2.3 Topology of the micro-structure 16

2.3.1 Detection of largest scales: autocorrelation 17

2.4 Synthetic porous geometry generator 18

2.4.1 Acicular mullite 19

3 Calculation of the flow field. **23**

3.1 Introduction 23

3.2 Theoretical background 26
3.2.1 Discretisation of Boltzmann equation 26
3.2.2 Numerical integration 29
3.2.3 Equilibrium distribution function 30
3.2.4 The collision operator 33
3.2.5 Chapman-Enskog expansion 35
3.2.6 Scaling Navier-Stokes to lattice units 39
3.2.7 Summary of hypothesis 41
3.2.8 Lattice Boltzmann *vs.* Classical CFD 42
3.3 Boundary Conditions 44
3.3.1 Fixed wall: bounce-back 45
3.3.2 Non-slip wall. Inamuro boundary condition. 46
3.3.3 Zoe & He boundary conditions 47
3.3.4 Boundary Conditions as an optimisation process 48
3.3.5 High order boundary conditions 59
3.4 Grid refinement 59
3.5 Software 63
3.5.1 Test environment 63
3.5.2 LaBMoTer 64
3.5.3 waLBerla 65
3.6 Validation 66
3.6.1 3D driven-cavity 66
3.6.2 Circular cylinder in cross-flow 67

4 Particle modelling **73**

4.1 The Diesel particles 73
4.1.1 Formation process 73
4.1.2 Characterisation 74
4.2 Particle transport 76
4.2.1 Trajectory calculation 78
4.2.2 Drag force 79
4.2.3 Particle-Fluid interaction 82
4.2.4 Brownian motion 85
4.3 Computed generated particles. Ballistic collision. 86
4.3.1 Theoretical background 87
4.3.2 Algorithms: impact detection 97
4.3.3 Non-dimensional descriptors for populations 102
4.3.4 Simulations 104
4.3.5 Evolution of the number of agglomerates in space. Generating solid phase. 117
4.3.6 Conclusion 121

5 Filter simulations **125**

5.1 Simulation setup 126
5.1.1 Geometry generation 126
5.1.2 Detection of the smallest scales 127
5.1.3 Grid dependence study 131
5.1.4 Cross-correlations and autocorrelations for the microstructure and the velocity 131
5.1.5 Influence of the peripheral boundary conditions 134
5.2 Analysis of the results 137

6 Conclusions and future works **141**

6.1 Conclusions 141

6.1.1 Geometry generator 142

6.1.2 Fluid Solver: Lattice-Boltzmann Method 142

6.1.3 Optimum setup 144

6.1.4 Fluid field solution 144

6.1.5 Particle Motion 145

6.2 Future works 148

6.2.1 Solvers, models and algorithms 148

6.2.2 Extra simulations 152

A List of Symbols **155**

A.1 Acronyms 155

A.2 Mathematical symbols 156

List of Figures

1.1 Scheme of the exhaust duct in a typical automotive engine. . . 2

2.1 Creation process of the computer-generated mullite. 21

3.1 Flow regime *vs.* simulations 24
3.2 6D representation of LBM distribution 27
3.3 Elastic binary collision scheme. 33
3.4 Boundary condition scheme. 44
3.5 Bounce-back boundary condition 54
3.6 Free slip boundary condition 56
3.7 LBM refinement techniques 62
3.8 Dimensionless velocity components in 3D driven cavity 67
3.9 Flow field for an immersed circular cylinder 68

4.1 Formation process of Diesel particles. 75
4.2 Diesel particle distribution *vs.* aerodynamic diameter 77
4.3 Gaussian kernel 84
4.4 Validation for Brownian motion 85
4.5 Collision detection algorithm 101

4.6 Population of agglomerates at the end of a simulation. 105

4.7 Gyration radius *vs.* the number of precursors 106

4.8 Number of agglomerates *vs.* number of precursor per agglomerate 108

4.9 Probabilistic and cumulative density distributions of residuals . 109

4.10 Exponents of agglomerates decay 111

4.11 Number of agglomerates *vs.* time 112

4.12 Histograms of the residuals 115

4.13 Measured population *vs.* random generated populations 117

4.14 Detail of depositions 121

5.1 Measured fractal dimension 130

5.2 Mesh independence study 132

5.3 Cross-correlations of the phase and the velocity components . . 133

5.4 Boundary conditions setup 135

5.5 Logarithm of the energetic norm of the error. 136

5.6 Pressure drop *vs.* velocity across porous material 138

5.7 Results of an slide for 16kPa of pressure drop. 139

List of Tables

3.1 Even moments for centred Gaussian distribution 31

4.1 Parameters and their variations in the study. 104

4.2 Setup for particle generation 120

5.1 Parameters for grid independence study 131

Chapter 1

Introduction

1.1 Motivation and objectives

Contamination is becoming an important problem in great metropolitan areas. A large portion of the contaminants is emitted by the vehicle fleet. So much so that some of the largest cities in Europe such as London or Paris are limiting the circulation of personal cars in the city centre. Reducing the speed limit or forgiving the circulation of cars in alternating days depending on the license plate are two of the most usual measures. Other cities such as Madrid activates the anti-contamination protocol if the measured pollutants concentration excesses a healthy threshold. At European level, as well as in other economical areas, the regulation is becoming more and more restrictive. Euro regulations are the best example of this tendency.

Specially important are the emissions of nitrogen oxide (NO_x) and Particle Matter (PM) among others such as unburned hydrocarbons. Two different strategies exist to reduce the emission of pollutants. One of them is trying to avoid their creation. Modifying the combustion process by means of different fuel injection laws or controlling the air regeneration are the typical methods. The second set of strategies is focused on the contaminant elimination. The NO_x are reduced by means of catalysis and/or reducing atmosphere, usually created by injection of urea. The particle matter is eliminated using filters. The filter traps the particles forming a deposition on its walls. The filter is regenerated by means of delayed post-injections, burning and reducing the deposition.

Most of the strategies to reduce the emission of contaminants penalise fuel

consumption. The particle filter is not an exception. Its installation, located in the exhaust duct (see figure 1.1), restricts the pass of the air. It increases the pressure along the whole exhaust line before the filter. With higher pressure in the exhaust pipe the regeneration of the air in the cylinder is more difficult. With a turbo-group installed the expansion relation of the turbine is reduced, limiting the amount of energy available for the compressor. In both cases there is a penalty in performance. Optimising the filter is then an important task. The efficiency of the filter has to be good enough to obey the contaminant normative. At the same time the pressure drop has to be as low as possible to optimise fuel consumption and performance. The objective of the thesis is to find the relation between the micro-structure and the macroscopic properties. With this knowledge the optimisation of the micro-structure is possible.

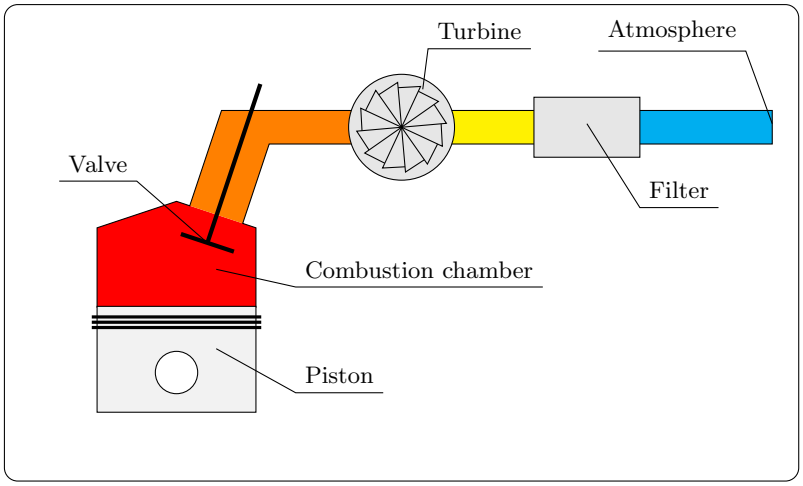

Figure 1.1: Scheme of the exhaust duct in a typical automotive engine.

1.2 Justification and previous work

This thesis is focused on the deposition and filtering study at micrometer scale. The physics at micro-scale level can explain the macroscopic properties of the material. Several accurate models of filters are based on these properties. One example is the model developed by Torregrosa *et al.* [1]. The architecture of the filter is described as a collection of ducts interconnected by porous material. Each duct is characterised by one-dimensional unsteady differential equation returning very accurate results. Consequently the model developed by Torregrosa *et al.* can be used to design the filter and to study its integration in the exhaust duct. Payri *et al.* [2] also develop a model to analyse the

integration of the filter. This model directly relates the macroscopic properties of the material with the pressure drop.

This thesis goes one step further in the level of detail. One of the objectives is to find the relation between the micro-structure descriptors and the macroscopic properties. The results of the simulations also return the distribution in space of the depositions and the flow field through the filter. The physical mechanisms can be explained in detail from these results. Understanding the physical mechanism is the first step to optimise the micro-structure. Several tools and solvers have been developed for this purpose.

The first and probably most important developed solver in the frame of this thesis is the Computational Fluid Dynamics (CFD) part. Taking into account the geometry to be simulated and the flow regime the best option to be considered is a Lattice Boltzmann Method (LBM). This kind of solvers have multiple advantages in comparison with classical CFD techniques. Previous research of other authors corroborates this.

Pan, Luo and Miller [3] apply several LBM techniques, boundary conditions and refinement levels to calculate beds of spheres. They compered the results to analyse their influence in the solution. This is a work to be done before any large simulation of porous media. The conclusions of this paper can be used to decide which collision operator is the most appropriate to get accurate solutions and maximise performance.

In the particular case of Diesel filters, several researchers simulate them. Hayashi and Kubo [4] calculate filters for two different scales. The largest scale is able to characterise the whole device. In these simulations the porous material is characterised by its macroscopic properties. The smallest scale simulates a fraction of porous wall. The porous medium is directly simulated from its micro-structures. Particle transport and deposition are also considered. Yamamoto and his team [5, 6, 7, 8, 9, 10, 11], from Nagoya University, simulate the physics of filters. They use LBM to solve the flow field. The micro-structure is measured from tomography of samples. Stewart *et al.* [12], from Northwest Pacific National Laboratory (NPNL) did an analogous work. They simulated the deposition of Diesel particles in a filter made of mullite. As well as Yamamoto's team, they obtained the geometry from tomography of samples. Fraunhofer Institute for Industrial Mathematics (ITWM) dedicate resources to develop solvers of porous media and other materials GeoDict [13]. They have a lot of simulations in fluid dynamics at micro-scale level. Some of their simulations are focus on the filtering process of Diesel particles [14, 15, 16]. As difference to previous researchers GeoDict creates the geometry instead of getting it from physical samples. They developed a software for chaotic procedural-generated geometries.

With respect to the suspended particles in the fluid, the most important parameter is the aerodynamic equivalent diameter. This magnitude is the easiest to be measured, reason why most of the research about characterising Diesel particles are focused on it. It explains the trajectory of the particles from an aerodynamic point of view. Some investigation about the relation between real geometry and descriptors can be found in the articles written by DeCarlo *et al.* [17]. However, due to the large amount of scales to be simulated, other physical phenomena have to be taken into account depending on the particle size. The trajectory of the smallest particles is influenced by Brownian motion (diffusion). This magnitude is correlated with the equivalent aerodynamic diameter but with large amounts of uncertainty. In the other end are the largest particles. The assumption of point particles is not valid for the biggest scales. The particle diameter can cover several cells. 3D effects have to be considered to detect collisions with the walls and to compute the trajectory correctly. Again, the size of the particle correlates with the aerodynamic diameter but with uncertainty. A new objective appears. The population of particles characterised by their equivalent aerodynamic diameter has to be converted to physical magnitudes. Uncertainty Quantification (UQ) techniques are used to convert the measurements in equivalent units to physical units. The process to find this relation is decomposed in two parts. The Diesel particles are very well described as agglomerates of spheres. The first part consists on create realistic agglomerates able to described statistically the properties of real Diesel particle populations. The second stage consist on measured the aerodynamic properties of the created agglomerates. This second part could not be finished on time and only the basic solver and its theory are explained in this document.

With respect to the particle Diesel, their shapes are not well investigated in detail. The fractal dimension is used to describe them. It measures how compact is an agglomerate. Some authors try to described the 3D structure from their image at the microscope and then calculate the fractal dimension [18]. Some others simulate the formation process. This last strategy is the followed in this thesis. A new solver has been developed to simulate the agglomeration process. It has been supposed free molecular regime. It means that the molecules travel large distance before collide with others in comparison with the characteristic length scale. The precursors of Diesel particles and the mean free path (mean distance between collisions) of air are of the same order of magnitude. The air can not be considered as a continuum media. On the contrary, it has been supposed ballistic agglomeration (no fluid, only particles). This assumption saves a lot of computational effort paying a slightly penalty in the accuracy of the results. Other researchers have simulated analogous ballistic agglomeration such as [19, 20, 21, 22, 23, 24, 25, 26]. However the developed software innovates in two different aspects. Firstly, High Performance Computing capabilities allow simulations with particles formed by several tens of thousands of precursors. Secondly a stochastic value decides when after an impact the colliding agglomerates fuse and when they simply

collide. Different scales of agglomerates present self-similarity. It means that the agglomerates are fractals. The fractal dimension of the resulting population has been calculated. Dekkers and Friedlander [26] measures the resultant aggregates for continuous, rarefied and free-molecular regimes.

The evolution of the particle population in time is also of interest. A dependency between the particle distribution and time can be established. A modified decay equation relates the number of agglomerates with time. This equation includes the parameters which control the agglomeration process. The relation is specified by studying the results of several combinations of these parameters. However there is a stochastic component in the solution. This component is a random variation with respect to the expected value. The uncertainty is characterised by UQ techniques. A generic population can be reconstructed from the controlling parameters and a random deviation. In other words, the same combination of input parameters can return multiple values.

With respect to the aerodynamic studies of the agglomerates. The diameter of the precursors (smallest Diesel particles) rounds 10nm. The distance between collisions of air molecules is the same order of magnitude. The hypothesis of air as a continuous medium is not valid to calculate the aerodynamic of agglomerates. A new CFD solver has been developed to simulate this kind of fluids. It is based on Direct Simulation Monte-Carlo (DSMC). This solver is the appropriate to simulate rarefied flows. DSMC is a Lagrangian method. Each molecule (or set of molecules) is calculated separately. They are sparse and the concentration is relatively low. The number of molecules contained in a single cell is not representative enough to get a smooth macroscopic field. Macroscopic magnitudes such as temperature, density or speed are recovered after collecting statistics for a while.

A solver based on DSMC has been developed. In addition to the physics several boundary conditions are implemented. Inlet and outlet are replaced by reservoirs just as other researchers do. For the wall new boundary conditions have been developed. The new boundary condition is calibrated from simulations of Couette flow. The drag of a sphere has been successfully calculated. Unfortunately the development time of the algorithm to simulate triangulate surfaces is too long, as well as the algorithm to convert a generic agglomerate into a triangulate surfaces. Even so the advances in these fields are also included in this thesis.

Once the particles are well characterised they can be included in the simulations. They are tracked computing their trajectory. Depending on the particle size the dominant physical phenomenon is different. The diameters of the smallest particles round ten nanometers. The particles with this size diffuse in the air. The discrete counterpart of the continuous Fick's law for diffusion is

Brownian motion. Brownian motion dominates the movement at nanomenter scales. The effects disappear when the particles become larger. Intermediate sizes are only affected by the aerodynamic properties of the particles and inertia. When the size of the particles is comparable to the lattice spacing three dimensional effects have to be considered. Point particle hypothesis is not valid any more. The agglomerate is affected by multiple nodes of the lattice instead of by single one. In addition the particle momentum is the same order magnitude large than the characteristic momentum of a single node. Under these conditions the reaction force becomes significant. Particle tracking and flow field are coupled. The details of the particles can no be capture by the lattice. Their presence is modelled by a kernel function.

The deposition happens when a particle impinges the wall. During this process the particle stops and all its kinetic energy is dissipated. The probability of a particle to become part of the deposition depends on this magnitude. When the kinetic energy is large the probability to be part of the deposition is low and *vice versa*. The alternative to the deposition is the collision. It happens when the energy to be absorbed is too high. The restitution coefficient fixes the amount of remaining kinetic energy.

1.3 Methodology

1.3.1 Work philosophy

This thesis has been laid out as an optimisation problem. The variable to be maximised is the advance in the filtering simulation. In other words, it is wanted to simulated as much as possible physical phenomena.

As in every optimisation process there are restrictions. The thesis is not an exception. Computational resources, rigour of the results and development time are the most important ones. The algorithms to solve the different problems have to be adapted to the computer architecture used for the thesis, a shared memory machine with 2 Central Processing Unit (CPU)s with 6 cores each (24 threads in total with hyper-threading technology) and 128GB. This restricts the simulation in two different ways. Firstly each simulation can not consumes more than 128GB. This fact forces to create adaptive solvers. The memory is consumed by the regions of the domain where small details have to be capture instead of by the entire domain. Secondly the algorithms have to be as fast as possible to get results rapidly using whole the machine. High Performance Computing (HPC) techniques become mandatory. Some strategies like divide and conquer algorithms or vectorisation are widely used. The workstation is also equipped with a Graphics Processor Unit (GPU) for

General-Purpose computing on Graphics Processing Units (GPGPU). This device has been used to create a test environment with maximum performance. Due to its low memory capacity it can be only used for small computations. Its used accelerates the development process. With this technology the tests are solved in minutes, avoiding to wait for the results for a long time.

Other of the restrictions is rigour in the development and the results. Each assumption and hypothesis are carefully applied and demonstrated. Several models are applied if the range of validity of each hypothesis is not large enough to cover the entire populations. Mesh independent studies corroborates that the solutions of the simulations are not affected by the discretisation. These studies are extended to the domain size. Cross-correlations demonstrate how the domains of the simulations are big enough to keep inside multiple of the biggest structures. It makes the solution of these simulations representative enough to avoid the presence of some anomaly and obtain averaged results. About the results and proposed models, every one have been decomposed in their deterministic and stochastic parts. With respect to the deterministic component, over-fitting, one of the most usual errors in modelling, has been avoided using the appropriate statistical tools. UQ techniques also model the stochastic part. It is not usual to see this type of studies in other researches. The contribution of the stochastic component is specially important if the solution presents random variations. Such is the case of the relation particle size *vs.* number of precursors. The same number of precursors can return multiple gyration radius (an indirect measured of the particle size). It is needed to characterise the whole population instead of simply the expected value.

1.3.2 Work flow

The filtering process can be decompose in multiple physical phenomena.

1. Geometry generation or measurement
2. Domain discretisation
3. Solution of the flow field
4. Particle transport
5. Deposition
6. Regeneration

The first idea was to include in the simulation all the previous points except regeneration. However, due to some unexpected difficulties particle deposition can not be included in time.

The development of each part has been done in sequential way. The geometry generator was the first tool to be developed. It recreates a filter of mullite. It consists on multiple needles of crystal whose interstices filter the particle Diesel. The generator is based on random numbers. It is a special case of procedural generation.

After creating the solid phase is time to solve the flow field through it. The developed solver for such purpose is based on lattice Boltzmann techniques. The first task to do is to discretise the domain to apply the numerical method on it. Discretising the domain is closely related with the solver of the flow field. At the beginning uniform discretisation was used. However the required memory to solve the entire domain exceeds the available memory of the computer. In addition, solving the whole domain with fine lattice is computationally very expensive. A new solver was developed to calculate with non-uniform adaptive lattices. At the same time the mesher was also developed. This software is able to calculate the flow field changing the discretisation dynamically to adapt the mesh to the solution. However it has a penalty in performance. The computational cost per cell increases in more than an order of magnitude. The framework waLBerla, developed by Lehrstuhl für Systemsimulation (LSS), a department of Friedrich-Alexander Universität Erlangen-Nürnberg (FAU), combines good performance with multi-resolution mesh. Some of the simulations have been done in collaboration with this department and using their framework.

Once the flow field is calculated and the solution is stationary it is time to include particles. The simplest model to calculate trajectories and consequently the first to be implemented is inertia-free motion. This hypothesis is valid when the drag force is much larger than the inertia of the particle. The trajectory is coincident with the trajectory of a fluid particle. For stationary non-turbulent flows trajectory and pathline are coincident. Consequently the trajectory of the particles are approximated by the path-line.

The problem of this model is its poor range of validity. The inertial effects for most of the particles are not negligible. Both drag force and inertia have to be considered. The particles tends to follow the pathlines except for abrupt curvatures, when the centrifugal forces push the particles far from the pathline. Most of the particle matter, considering the distribution in mass, are in the validity range of this less restrictive hypothesis. This is in fact the most widely used model to track Diesel particles. Most of the measurements, usually based on equilibrium between drag force and external forces such as gravity, centrifugal forces or electromagnetic forces, use normalised magnitudes. This decouples the particular properties of each particle such as shape or density, with high variability, from the aerodynamic behaviour. To reduce the number of variables to take into account the measured particles are described in terms of their equivalent magnitudes. Equivalent magnitudes come from equivalent

particles. They have spherical shape and normalised density of 1000Kg/m^3. The diameter of the equivalent spherical particles is called equivalent aerodynamic diameter. Their motion of equivalent and original particles are identical. Calculating the trajectory of the equivalent particle returns the original in the range of validity of the hypothesis. When the magnitude of interest is particle matter, assuming this hypothesis returns accurate results. On the contrary, if the magnitude of interest is the number of particles instead of the mass, Brownian motion has to be considered. Brownian motion is a random movement of small objects. It is the Lagrangian version of the Fick's law for diffusion. In the case of interest, most of the particles are affected by this phenomenon.

In the opposite limit are the largest particles. When the particle size is comparable to lattice spacing the particle can not be considered as a point. 3D effects have to be included. In this thesis the 3D effects are simplified and modelled by a Kernel Gaussian function. Due to the small momentum quantity trajectory calculation and flow field can be decoupled without penalty in accuracy. For stationary flow field without any variation in time trajectory calculation is just a simple post-processing task. However, when this hypothesis is not valid, the momentum exchange between the particle and the surrounding nodes is not negligible. The fluid pulls the particle and the particle pushes the fluid. The flow patterns can be affected by the presence of the particles and consequently both particle motion and flow field have to be calculated together in coupled way.

Assuming the equivalent diameter to calculate Brownian motion can result in errors in the trajectory. It is then necessary knowing the real properties of the particles, at less in the range affected by Brownian motion. The measurements are usually done in equivalent magnitudes instead of real ones. Knowing the relation between equivalent magnitudes and real magnitudes becomes mandatory. For such purpose two solvers are developed. The first solver recreates the formation of agglomerates (particle Diesel) by ballistic collisions. It returns realistic geometry. The second solver calculates the drag force (drag coefficient) of the generated agglomerates. Then the equivalent particles are calculated. At this point there are as much couples of real and equivalent particles as simulations. Using statistics a model to relate both magnitudes can be created. It allows to convert the equivalent population of particles into a real population. Unfortunately this part has not been finished on time. However the methodology are described in detail for future works.

1.4 Resources and tools

Most of the simulations were run in a workstation. The workstation used for the simulations has two CPUs Intel Xeon ES-2640 at 2.50GHz, with 6

cores each and hyper-threading technology, 8 slots of DDR3 RAM at 1333MHz with 16GB each to sum 128GB in total, and a nVidia Tesla C2075 with 448 CUDA cores at 1.15GHz with 6GB DDR5 RAM memory. Every developed application has been optimised to run in this computer. Parallelisation has been usually done using Open Multi-Processing (OpenMP) library, the best option for shared memory machines. Special algorithms, such as divide and conquer strategies, reduce the computational load to improve performance and save simulation time. HPC techniques have also been implemented for the same purpose. The limiting factor is Random Access Memory (RAM) memory. Adaptive discretisation of the flow field to the geometry and the solution reduces the number of required cells without lost of accuracy. However refinement has a negative effect in performance. The memory is not aligned. Their location can not be known *a priori*. The access to the registers is not direct. Indexing is needed. As consequence there are much more cache misses, prefetching is ineffective, algebra can not be vectorised and indexes for linking have to be calculated each iteration.

Some of the solvers and subroutines have been implemented for the GPU. These implementations take advantage of the great amount of cores contained on it to solve the same problem in a fraction of time. The difference in computational time for the most vectorizable applications could be more than two orders of magnitude in comparison with the same calculation in sequential mode by a CPU.

Some other simulations have been done in collaboration with LSS using their framework waLBerla. This software is optimised to run in the most powerful supercomputers. The architecture of this type of machines is usually distribute memory. The cluster of LSS and SuperMUC supercomputer are of this kind. Both of them have been used to run some simulations of the porous media. Typical simulations are launched in the cluster, whereas the largest ones have been calculated by SuperMUC.

Several programming languages have been used depending on the purposed of the developed software. The solvers, developed with HPC strategies, are almost written in C++ and Fortran. OpenMP, MPI and CUDA are the libraries used for parallelisation. When the variable to be optimised is the development time instead of the software performance the code was developed in high programming languages. Scripts are usually developed in Linux Bash or Python. About basic tests and auxiliary tools, they are usually written in Python or Matlab/Octave.

The post-processing tasks are usually done with the specific software Paraview. This software is a very powerful tool. It includes several filters for different purposes: results visualisation, data extraction, time regression, interpolation, etc. Some macros have been developed *ad hoc* in Python to effi-

ciently analyse the results of the simulation. The data of interest is automatically extracted with the macros. A second option whose single purpose is data visualisation is Blender. Blender is a software specialised in animation. One of its more powerful tools is its render engine. This engine is able to render volumetric textures based on voxels. The results of the simulations are converted to volumetric textures and then rendered with this software. The whole process is done by macros written in python. Stream-lines and particles are also imported to Blender by other macros. The code is able to generate the geometry of the path-lines and fix the properties of the material for rendering. The colour and optionally the radius along the path-line depends on a variable of the solution.

Bibliography

[1] Torregrosa, A.J.; Serrano, J.R.; Arnau, F.J.; Piqueras, P. A fluid dynamic model for unsteady compressible flow in wall-flow diesel particulate filters. *Energy*, 36(1):(2011), 671–684.

[2] Payri, F.; Broatch, A.; Serrano, J.R.; Piqueras, P. Experimental–theoretical methodology for determination of inertial pressure drop distribution and pore structure properties in wall-flow diesel particulate filters (dpfs). *Energy*, 36(12):(2011), 6731–6744.

[3] Pan, C.; Luo, L.S.; Miller, C.T. An evaluation of lattice boltzmann schemes for porous medium flow simulation. *COMPUTERS & FLUIDS*, 35(8-9):(2006), 898–909. 1st International Conference for Mesoscopic Methods in Engineering and Science (ICMMES), Tech Univ Braunschweig, Braunschweig, GERMANY, JUL 25-30, 2004-2005.

[4] Hayashi, H.; Kubo, S. Computer simulation study on filtration of soot particles in diesel particulate filter. *Computers & Mathematics with Applications*, 55(7):(2008), 1450 – 1460. Mesoscopic Methods in Engineering and Science.

[5] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Lattice boltzmann simulation on porous structure and soot accumulation. *Mathematics and Computers in Simulation*, 72(2-6):(2006), 257–263.

[6] Yamamoto, K.; Satake, S.; Yamashita, H. Lattice boltzmann simulation on flow with soot accumulation in diesel particulate filter. *International Journal of Modern Physics C*, 18(4):(2007), 528–535.

[7] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Fluid simulation and x-ray ct images for soot deposition in a diesel filter. *European Physical Journal-Special Topics*, 171:(2009), 205–212.

[8] Yamamoto, K.; Oohori, S.; Yamashita, H.; Daido, S. Simulation on soot deposition and combustion in diesel particulate filter. *Proceedings of the Combustion Institute*, 32:(2009), 1965–1972.

[9] Yamamoto, K.; Yamauchi, K.; Takada, N.; Misawa, M.; Furutani, H.; Shinozaki, O. Lattice boltzmann simulation on continuously regenerating diesel filter. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 369(1945):(2011), 2584–2591.

[10] Yamamoto, K.; Yamauchi, K. Numerical simulation of continuously regenerating diesel particulate filter. *Proceedings of the Combustion Institute*, 34:(2013), 3083–3090.

[11] Yamamoto, K.; Ohori, S. Simulations on flow and soot deposition in diesel particulate filters. *International Journal of Engine Research*, 14(4):(2013), 333–340.

[12] Stewart, M.L.; Ward, A.L.; Rector, D.R. A study of pore geometry effects on anisotropy in hydraulic permeability using the lattice-boltzmann method. *ADVANCES IN WATER RESOURCES*, 29(9):(2006), 1328–1340.

[13] Fraunhofer Institute for Industrial Mathematics. GeoDict (2017).

[14] Rief, S.; Kehrwald, D.; SCHMDT, K.; Wiegmann, A. Fraunhofer software tools geodict/filterdict for the simulation of diesel particulate filters. *NAFEM: Reliable Use of Numerical Methods in Upfront Simulations. Wiesbaden, Germany.*

[15] Rief, S.; Schmidt, D.M.K.; Wiegmann, A.; Transportations, T. Virtual diesel particulate filters: Simulation of the structure, exhaust gas flow and particle deposition (2009).

[16] Wiegmann, A.; Rief, S.; Kehrwald, D. Computational study of pressure drop dependence on pleat shape and filter media. In *Filtech 2007 International Conference for Filtration and Separation Technology*, volume 1, pages 79–86 (2007).

[17] DeCarlo, P.F.; Slowik, J.G.; Worsnop, D.R.; Davidovits, P.; Jimenez, J.L. Particle morphology and density characterization by combined mobility and aerodynamic diameter measurements. part 1: Theory. *Aerosol Science and Technology*, 38(12):(2004), 1185–1205.

[18] Park, K.; Kittelson, D.B.; McMurry, P.H. Structural properties of diesel exhaust particles measured by transmission electron microscopy (tem): Relationships to particle mass and mobility. *Aerosol Science and Technology*, 38(9):(2004), 881–889.

[19] Sorensen, C.M. The mobility of fractal aggregates: A review. *Aerosol Science and Technology*, 45(7):(2011), 755–769. Cited By 105.

[20] Wang, G.M.; Sorensen, C.M. Diffusive mobility of fractal aggregates over the entire knudsen number range. *Phys. Rev. E*, 60(3):(1999), 3036–3044.

[21] Eggersdorfer, M.L.; Pratsinis, S.E. The structure of agglomerates consisting of polydisperse particles. *Aerosol Science and Technology*, 46(3):(2012), 347–353. Cited By 32.

[22] Eggersdorfer, M.L.; Pratsinis, S.E. Agglomerates and aggregates of nanoparticles made in the gas phase. *Advanced Powder Technology*, 25(1):(2014), 71–90. Cited By 28.

[23] Goudeli, E.; Eggersdorfer, M.L.; Pratsinis, S.E. Coagulation-agglomeration of fractal-like particles: Structure and self-preserving size distribution. *LANGMUIR*, 31(4):(2015), 1320–1327.

[24] Goudeli, E. and Eggersdorfer, M. L. and Pratsinis, S. E. title. Coagulation of agglomerates consisting of polydisperse primary particles. *LANGMUIR*, 32(36):(2016), 9276–9285. PMID: 27536889.

[25] Morgan, N.; Kraft, M.; Balthasar, M.; Wong, D.; Frenklach, M.; Mitchell, P. Numerical simulations of soot aggregation in premixed laminar flames. *Proceedings of the Combustion Institute*, 31(1):(2007), 693–700.

[26] Dekkers, P.J.; Friedlander, S.K. The self-preserving size distribution theory. *Journal of Colloid and Interface Science*, 248(2):(2002), 295–305.

Chapter 2

The porous media and its nature

Porous media are very common in nature and its use is very extended in engineering. Filtering is one of their applications. In the following sections it is briefly explained the filtering process, the micro-structure of the material, its mathematical characterisation and how to create computer-generated porous media with procedural generation techniques.

2.1 Description

The most important hydrodynamic characteristic of porous media is permeability. Permeability explains the hydrodynamic behaviour of the material from a macroscopic point of view. When the flow at micro-scale level is low the pressure drop scales linearly with the flow rate. This relation is known as the Darcy's law

$$\vec{q} = -\frac{1}{\mu}[\kappa]\vec{\nabla}P. \tag{2.1}$$

$\vec{q}$ is the flow rate, $[\kappa]$ is the permeability tensor, μ is the viscosity and P is pressure. There are several variations of Darcy's law to take into account other effects not covered by the hypothesis. The Forchheimer variation includes an inertial term to take into account non-linear effects. The modified equation along the pressure gradient direction is

$$\frac{\partial P}{\partial x} = -\frac{\kappa}{\mu}q - \frac{\rho}{\kappa_1}q^2. \tag{2.2}$$

The new term includes the density of the fluid ρ and the inertial permeability κ_1. For isotropic materials the permeability tensor can be replaced by a scalar κ. The flux is related with the fluid velocity $\vec{u}$ by the porosity ϕ

$$\vec{u} = \frac{\vec{q}}{\phi}. \tag{2.3}$$

The porosity is the relation between the volume covered by the fluid phase over the total volume. Its value is in the range between 0 and 1. Each material is characterised by different values of porosity ϕ and permeability $[\kappa]$.

2.2 The filtering process

The filter traps particle matter in its interior creating deposits. At the beginning there are only two phases, the solid phase and the fluid phase. Particle matter deposition is the third phase. At micro-scale level this material is modelled as porous media. There is a porous material inside a porous material. The difference is in the scale. Whereas the studied geometry has a porous size of several microns the depositions have porous size in the order of tens of nanometers. The discretisation of the simulations of this thesis can capture details until 1μm. Lower scales have to be modelled by the Darcy's law 2.1.

The depositions constrict the pass of air through the filter. The mass flow rate is reduced for constant pressure gradients or the pressure drop increases for constant mass flow rates. The blocking process of the filters can be decomposed in two stages depending on the occlusion type. At the beginning the filter is clean. The particle matter starts to travel in the interior of the filter creating the first deposits. The quantity of deposited matter decreases with the depth. A lot of particle matter is also captured by the surface in contact with the dirty fluid. The deposits grow at the same time that they block the pass of the air. At the end of the first stage the particle matter obstructs every pore of the surface. All the air which crosses the filter crosses the deposits. Here starts the second stage. The suspended particles can not cross the deposits and precipitate on the surface. This layer grows to create a region between the solid phase and the fluid phase. This bed of precipitated matter is in practise a new filter over a filter, but with much smaller porous size.

2.3 Topology of the micro-structure

A lot of different materials can be used to made a Diesel filter. Adler summarises in his article [1] the filtering properties of several ceramic materials.

Every material crystallises into its own micro-structure. Accordingly to this criterion the filters can be characterised as foam filters, particulate filters, fibre filters, acicular filters, amorphous filters, *etc.* All the resulting geometries can be decomposed mathematically into its stochastic and deterministic parts. The proportion of each part describes how is the material. Autocorrelation is the magnitude used in this thesis to detect determinism. Values far from zero at large distances quantifies the deterministic component. The normalised autocorrelation of woven filters is a cosine curve. The maximum value at infinite distances is close to one. It means that the micro-structure of woven materials is almost deterministic. On the contrary, the autocorrelation values of chaotic porous media becomes zero after a distance.

2.3.1 Detection of largest scales: autocorrelation

The cross-correlation between two dependent variables f and g measures their similarity as a function of the relative displacement $\Delta\vec{\xi}$ in the $\mathbb{R}^D$ space along the independent variables ξ_d. It is defined as

$$(f \star g)\left(\Delta\vec{\xi}\right) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\vec{\xi}) g(\vec{\xi} + \Delta\vec{\xi})\, d\xi_1 d\xi_2 \cdots d\xi_D. \tag{2.4}$$

The discrete equivalence is

$$(f \star g)_{n_1,\cdots,n_D} = \sum_{m_1,\cdots,m_D=-\infty}^{\infty} f_{m_1,\cdots,m_D}\, g_{m_1+n_1,\cdots,m_D+n_D}. \tag{2.5}$$

The autocorrelation is the cross-correlation of a variable with itself $\mathscr{R}_f(\Delta\vec{\xi}) = (f \star f)$. It can be observed how the autocorrelation for zero displacement is the square of the euclidean norm of the function. Every dependent variable can be normalised dividing by the square root of the autocorrelation at zero displacement

$$\bar{f}(\xi) = f(\xi)/\sqrt{\mathscr{R}_f(0)}. \tag{2.6}$$

Any inner product of two normalised dependent variables is always bounded by the interval [-1, 1]. Because the autocorrelation for 0 displacement of a generic function is its norm, the result is one if the function is normalised. The obvious consequence is that the autocorrelation for 0 displacement is a global maximum. This strategy can be also applied to calculate the cross-correlation. Using normalised variables the result is again bounded by the interval [-1, 1]. However the maximum value is usually lower than 1 and its location is not necessarily at zero.

For N_d samples along the d-th dimension the computational cost of calculating the cross-correlation has order $\mathcal{O}(N_d^2)$. However the cost can be drastically

reduced if the signal is periodic along the d-th dimension. When the generic function f is periodic along ξ_d, then Fast Fourier Transforms (FFT) can be applied instead of the brute force algorithm. It reduces the order of the computation to $\mathcal{O}(N_d \log(N_d))$. If $\hat{f}(\kappa_d)$ is the function $f(\xi_d)$ in frequency domain for a generic wave number κ_d, then the cross-correlation of two dependent variables in frequency domain can be expressed as

$$\widehat{(f \star g)}(\kappa_d) = \hat{f}(\kappa_d)\hat{g}^*(\kappa_d), \tag{2.7}$$

where * denotes the complex conjugate value and $\hat{}$ means the FFT transform. Then applying the Inverse Fast Fourier Transforms (IFFT) the cross-correlation in the physical domain is returned.

Any coherent structure correlates with itself. The autocorrelation can be used to measure the size of these structures. To analyse the geometry using this technique a phase field ϕ is created for each material. Volumes covered by the associated material returns a phase field equal to 1, 0 otherwise. The cross-correlation between phase fields returns the intersection volume for every displacement. However these magnitudes can not be directly used to detect the structure size. Structure size detection requires normalised fields. The mean value has to be subtracted and the result divided by the square root of the autocorrelation at zero displacement.

The structure appears in the autocorrelation as an elliptic region around the origin. The shape of the correlated region provides information about the isotropy of the material and the orientation of its structures. Circular regions means that the material is isotropic. On the contrary, ellipses with high eccentricity are the autocorrelations of anisotropic materials. The axes of the ellipse are aligned with the principal axes of the largest coherent structures. The correlated region doubles the size of the coherent structure.

2.4 Synthetic porous geometry generator

The solid phase of a porous media is usually very complex. Because the little size of the structures the measurement of the geometry requires of very expensive techniques. One of them and, in fact, the most usual, is the tomography. Yamamoto's team [2, 3, 4, 5, 6, 7, 8] uses this technique to obtain the 3D image of the porous media structured in voxels. This returns an acceptable resolution (1μm) and covers acceptable domain sizes (400μm). However this is sometimes not good enough (see sections 5.1.2 and 5.1.3 of this document) and the domain could not be representative (see section 5.1.4 of this document).

When the experimental data is not reachable computer-generated solid phase can replace it. The software GeoDict [9] creates virtual filters for

a large set of materials [10, 11]. The micro-structure of the filter has been created following a similar procedure. In the following lines it is explained a set of techniques to create several types of computer-generated filters.

2.4.1 Acicular mullite

The filters made of acicular mullite are usually used in competitions. Thanks to its high porosity their pressure drop is very low compared with other types of filters for the same filtering efficiency.

The acicular mullite is manufactured from a precursor material block. Applying heat and using catalysis the precursor crystallises. The crystals are not randomly distributed. They grow up from specific locations called nucleation points forming radial aggregates.

The program which mimics this micro-structure uses populations of parameters to control the shape of the crystals, the location of the nucleation points, the number of aggregates per unit volume and the distribution of crystals in a single aggregate.

The first step in the creation process is to find the location of the nucleation points. Their existence is limited to the space covered by the block of the precursor material. They are supposed to be uniformly distributed along this region. Once the location of the nucleation point of each aggregate is fixed a set of crystals is created. How many crystals each aggregate contains is controlled by a random sample whose population follows a log-normal distribution. Log-normal distribution forces the number of crystals to be positive. After fixing the number of crystals each crystal has to be created. The creation process starts from a normalised crystal whose length and thickness are well known. Each one is a parallelepiped finished by two pyramids with fixed angle at its bases. The origin of coordinates coincides with its centre of gravity and the length follows the X direction of the local coordinate system. Then, two scales are applied along and across the crystal respectively. The first scale to be applied is for the length. The scale factor is again a sample coming from a log-normal distribution and it scales all dimensions. The second scale only modifies the crystal thickness and it is applied after the first scaling. That is, the second scale is not the thickness itself, it is the inverse of aspect ratio. This strategy avoids the existence of strange shaped crystals (extremely long or extremely thick). Again the aspect ratio is not always the same. It is different for each crystal and the population follows a log-normal distribution. With this two simple operations the crystal is correctly shaped. Now it has to be oriented and located. The last operation to be applied on the crystal is a translation to force the coincidence between the nucleation point of the aggregate and the origin of the local coordinate system associated to the crystal. At this

moment the centre of coordinates of the local system overlaps the centre of gravity of the crystal. That is, the geometry of the crystal is anti-symmetric respect to the local coordinate system. To destroy this property the crystal is translated along its main axis. The translated distance is called eccentricity of the crystal. This parameter is defined as the distance between the centre of gravity of the crystal and the nucleation point of the aggregate over the total length of the crystal. It is also random-generated and it is extracted from a normal population centred in 0. A sample can never exceed the value 0.5 so the standard deviation which describes the population has to be controlled. Otherwise the crystal could be far away from the aggregate. Now the needle points out the local X axis. However they grow up in every direction, in uniform way per steradian. The crystal has to be oriented in elevation and azimuth. Respect to the elevation, it can take values between $-\pi/2$ and $\pi/2$ [1]. A uniform random population is generated and transformed afterword to avoid an over-concentration of crystals around nadir and zenith. The operator arcsin is applied on every sample. Once the crystal is elevated it has to be rotated around the Z axis. This time the rotation angle can be extracted directly from a uniform random population after scaling its value by 2π. The final step is just to move the centre of coordinates of the crystal to the nucleation point. Whole the process is repeated for each crystal till complete each aggregate. The figure 2.1 resumes the process.

[1] Because the eccentricity is equally distributed to positive and negative values the elevation range can be reduced to the interval bounded by the limits 0 and $\pi/2$.

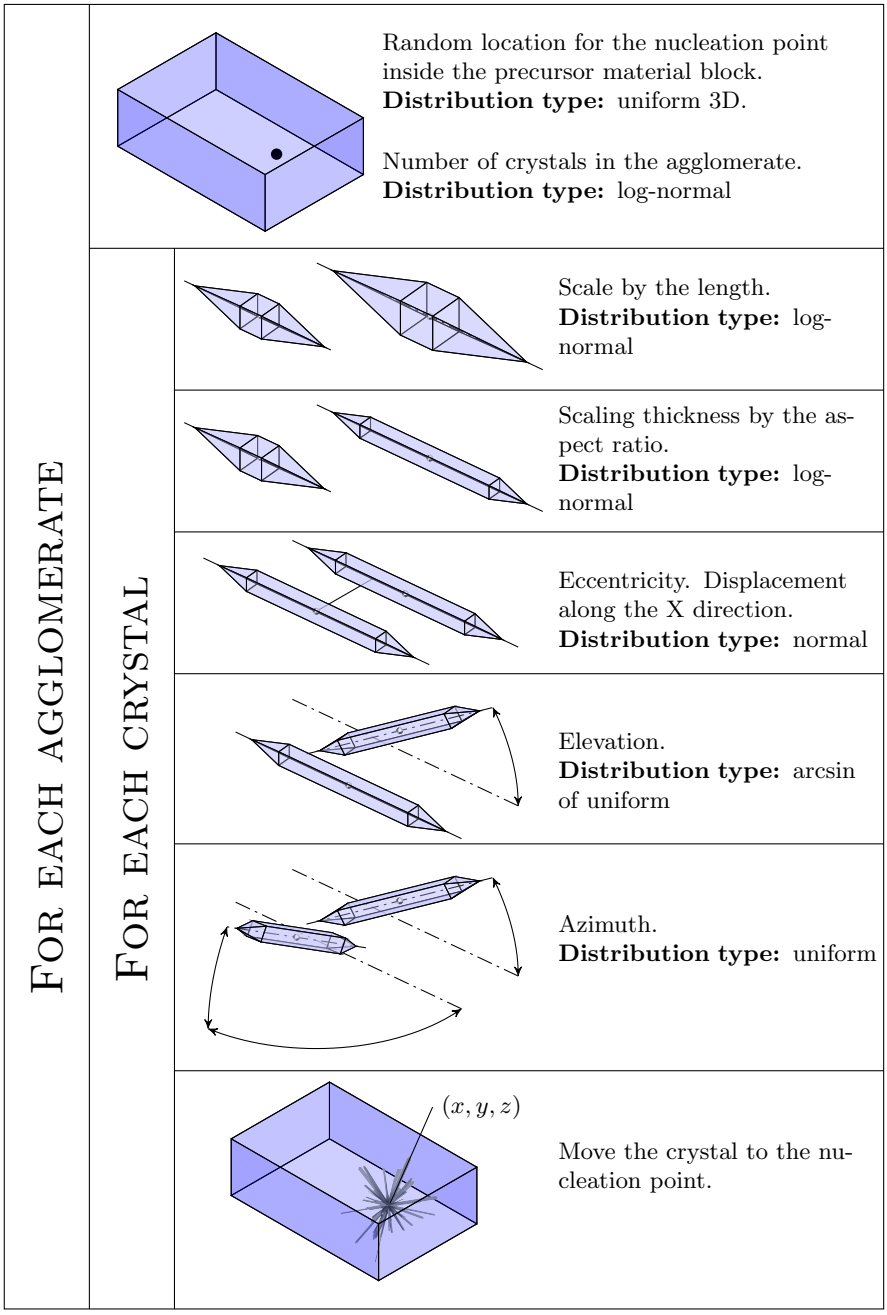

Figure 2.1: Creation process of the computer-generated mullite.

Bibliography

[1] Adler, J. Ceramic diesel particulate filters. *International Journal of Applied Ceramic Technology*, 2(6):(2005), 429–439.

[2] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Lattice boltzmann simulation on porous structure and soot accumulation. *Mathematics and Computers in Simulation*, 72(2-6):(2006), 257–263.

[3] Yamamoto, K.; Satake, S.; Yamashita, H. Lattice boltzmann simulation on flow with soot accumulation in diesel particulate filter. *International Journal of Modern Physics C*, 18(4):(2007), 528–535.

[4] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Fluid simulation and x-ray ct images for soot deposition in a diesel filter. *European Physical Journal-Special Topics*, 171:(2009), 205–212.

[5] Yamamoto, K.; Oohori, S.; Yamashita, H.; Daido, S. Simulation on soot deposition and combustion in diesel particulate filter. *Proceedings of the Combustion Institute*, 32:(2009), 1965–1972.

[6] Yamamoto, K.; Yamauchi, K.; Takada, N.; Misawa, M.; Furutani, H.; Shinozaki, O. Lattice boltzmann simulation on continuously regenerating diesel filter. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 369(1945):(2011), 2584–2591.

[7] Yamamoto, K.; Yamauchi, K. Numerical simulation of continuously regenerating diesel particulate filter. *Proceedings of the Combustion Institute*, 34:(2013), 3083–3090.

[8] Yamamoto, K.; Ohori, S. Simulations on flow and soot deposition in diesel particulate filters. *International Journal of Engine Research*, 14(4):(2013), 333–340.

[9] Fraunhofer Institute for Industrial Mathematics. GeoDict (2017).

[10] Rief, S.; Kehrwald, D.; SCHMDT, K.; Wiegmann, A. Fraunhofer software tools geodict/filterdict for the simulation of diesel particulate filters. *NAFEM: Reliable Use of Numerical Methods in Upfront Simulations. Wiesbaden, Germany.*

[11] Rief, S.; Schmidt, D.M.K.; Wiegmann, A.; Transportations, T. Virtual diesel particulate filters: Simulation of the structure, exhaust gas flow and particle deposition (2009).

Chapter 3

Calculation of the flow field. The Lattice Boltzmann Method

3.1 Introduction

The LBM is a mathematical artifice which emulates the physics of Navier-Stokes (NS) equations. It is based on the kinetic theory formulation, which relates the molecule motion with the macroscopic properties of the fluid. The solver to be applied depends on the scale. When the scale is very small the fluid is rarefied and the domain contains a reduced set of molecules which collide and interact with each other. Simulators based on Molecular Dynamics (MD) can emulate the mechanic of this type of fluids. The number of molecules contained in the domain increases rapidly with the density or the scale. When the concentration is large enough the hypothesis of continuous media becomes valid. Simulating the motion of every particle separately becomes unreachable in this context. Another approach is needed. The particle population contained in a differential of volume (continuous media permits this assumption) is statistically described by a three-dimensional distribution in speed. This distribution, which varies in space and time, depends on a reduced set of parameters. The distribution intrinsically also contains the values of the macroscopic magnitudes (typical magnitudes in classical fluid dynamics, such as density, pressure or velocity). As an example. The addition of the mass of every particle (integral of the population along the particle speed and volume) returns the mass of fluid contained in that volume. In analogous way, the sum-

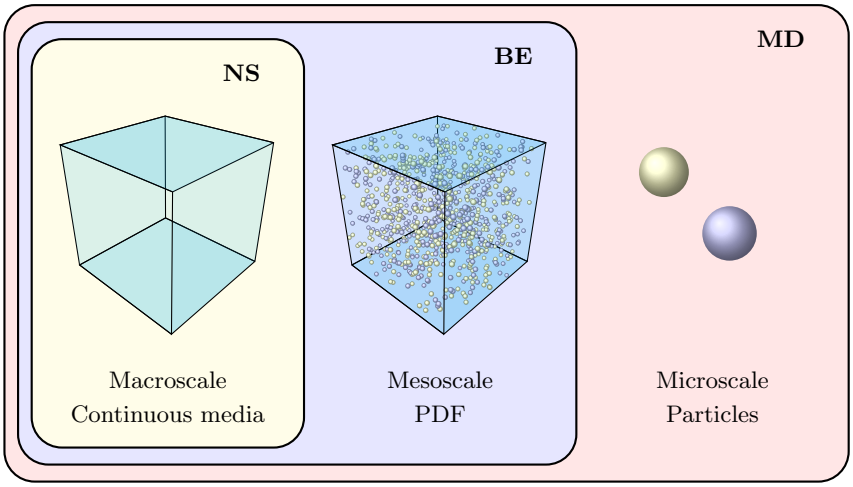

Figure 3.1: Flow regime *vs.* simulations

mation of all the linear momenta (mass of each particle times its speed) along the volume is the macroscopic momentum contained in the volume. Other magnitudes such as pressure, temperature or viscosity can also been obtained by more complicated ways. The equation which governs the evolution in time and space of particle populations is the Boltzmann equation. This equation is the base of the LBM, reason why it takes this name. The Boltzmann equation can solve what is called mesoscale. At mesoscales, the flow field is considered continuous but still rarefied. In other words, the equilibrium distribution of particles is not reached at each point.

When the characteristic time and length to achieve equilibrium is much smaller than the scale of the simulation, NS can be directly applied. This scale is called the macroscopic scale and it is the classical formulation of fluid mechanics. Because the large scales contain the smallest ones, the integral of the smallest scales returns the equation for a single point in largest scales. Thus the Boltzmann equation is obtained after integrating the MD equation along large time and space increments, and similarly NS is recovered after integrating the Bolzmann-equation. The figure 3.1 represents how each large scale is a simplification of the previous one. What Lattice Boltzmann Method does is to take advantage of this equivalence and replace the physical fluid by an equivalent rarefied gas described by means of artificial particle populations. The way how this gas tends to equilibrium emulates the effects of the viscosity in the real fluid. The macroscopic magnitudes such as velocity or pressure are implicit in the populations and they can be calculated easily. With pressure and velocity replaced by particle populations, the pressure correction loop is not needed anymore, saving computational effort.

The distributions, continuous in Boltzmann equation, are discrete in LBM. The discrete velocities have to be coherent with the domain decomposition. During a time step the particles can only travel from one node to other, so the discrete velocities have to be the distance between contiguous nodes over the time step. The decomposition, for the simplest cases, is uniform and structured, reason why the method includes the word "lattice" in its name. This condition is mandatory to keep the same set of discrete velocities everywhere. The lattice is immersed into the domain, it does not decompose it in strict sense. In other words, the lattice is not adapted to the geometry. The presence of BCs in generic locations is taken into account in the formulation instead of being part of the decomposition process. This is a direct consequence of the uniform-spaced restriction.

But the most distinctive characteristic of LBM is its explicit formulation. The use of particle distributions instead of fluid magnitudes gives this important advantage. The explicit formulation is local, the state of one node is only affected by the surroundings. From the computational point of view it makes the method highly parallelisable. It is a good candidate to be implemented for distribute memory machines and accelerators. On the contrary, implicit methods are global. They usually require the inversion of large matrices with the corresponding computational cost in numerical operations and communication. One more advantage of LBM is the speed of convergence, second order for the simplest schemes.

On the contrary LBM shows some negative counterparts. As every explicit method, LBM requires more iterations to converge. Consequently the method is not the most appropriate for small simulations in general. Other disadvantage is the memory consumption. Discrete distributions usually demand more space than macroscopic magnitudes. Finally, one more aspect to take into account is the time dependence. LBM is in origin non-steady, time is involved for every simulation.

What is mentioned above are the characteristics of the simplest LBM algorithms. However there are some advance LBM implementations which improve the previous exposed characteristics. For instance, one of the most extended improvements is the used of several lattices of different size making the decomposition adaptable to the geometry or the solution.

As direct consequence of LBM characteristics the best candidates to be simulated are:

- Complex geometries. The lattice is immersed into the domain, it does not decompose it. Because the mesh has not to be adapted to the geometry the meshing process is fast ans simple. This is the case of porous media, capillary vessels, riverbeds and any other chaotic geometry.

- Moving BCs. The immersed lattice makes LBM a good candidate to simulate moving geometry too. The mesh has not to be recomputed for every time step. Immiscible multiphase flows, deposition, crystal growth, deformable structures... are perfect candidates to be solved with LBM.
- Distribute memory machines and accelerators. The locality of the method brings the possibility of developing highly parallel applications. This quality makes the LBM the perfect candidate to run large simulations in machines of this architecture.

In this chapter the LBM method will be explained in detail. The following section explains what LBM is, the mathematical background, the hypothesis to take into consideration, how to recover NS and how to convert the physical units to lattice units. Immediately after the BCs are explained. In LBM the BCs are generally under-restricted. Consequently the restriction imposed at the boundaries can be accomplished by infinite populations. A new point of view is introduced in the current thesis with respect to the Boundary Condition (BC). They are considered as restricted optimisation problems.

3.2 Theoretical background

The formulation of LBM is based on the Boltzmann equation

$$\partial_t f + \vec{c} \cdot \vec{\nabla}_{\vec{x}} f + \vec{a} \cdot \vec{\nabla}_{\vec{c}} f = \Omega. \tag{3.1}$$

In this context $\vec{c}$ is the speed of the particles, $\vec{a} = \vec{F}/m$ the acceleration of a particle with mass m due to the force $\vec{F}$, $f = f(t, \vec{x}, \vec{c})$ is the particle distribution with respect to the particle velocity $\vec{c}$ at the location $\vec{x}$ at time t, Ω is the collision operator responsible of the redistribution of f along $\vec{c}$, and the operands $\vec{\nabla}_{\vec{x}}$ and $\vec{\nabla}_{\vec{c}}$ are the gradients in space and velocity of the particles respectively.

For a single location and instant, f represents how the matter is distributed in velocity. In other words, the distribution adds three more independent variables, creating a six-dimensional hyperspace (see figure 3.2). If the equation has to be solved numerically, these three extra dimensions have to be also discretised.

3.2.1 Discretisation of Boltzmann equation

There are three sets of independent variables to be discretised: time, space and velocity. The computational scheme is governed by the velocity discretisation.

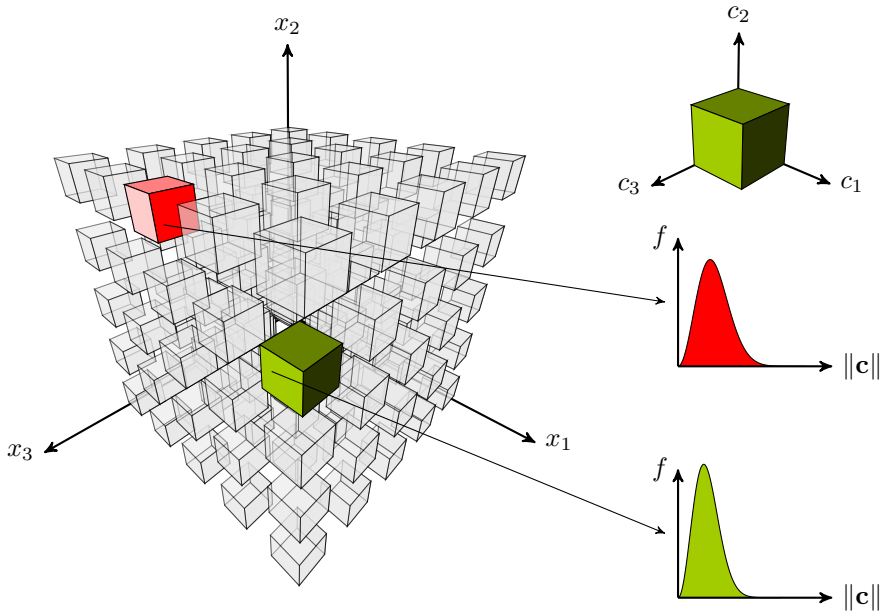

Figure 3.2: 6D-Hypercube representing the distribution of f. Each little cube represents the velocity domain. One of this exists for each location in space domain.

The nomenclature of the LBM scheme is normalised accordingly as DXQY, where X is the number of dimensions of the scheme and Y the number of discrete velocities. For example, the scheme to solve 3D domains with 19 discrete velocities is named D3Q19.

Every generic function $g(\vec{c})$ dependent on the velocity can be decomposed in orthogonal polynomials as

$$g(\vec{c}) = \sum_{\alpha} g(\vec{c}_\alpha)\mathscr{P}_\alpha(\vec{c})W(\vec{c}). \tag{3.2}$$

Where the subscript $_\alpha$ means "relative to the α-th discrete velocity", $g(\vec{c}_\alpha) = g_\alpha$ is the generic function evaluated in $\vec{c}_\alpha$, $\mathscr{P}_\alpha$ is the α-th polynomial of the orthogonal base, and $W(\vec{c})$ is the weighting function. The polynomial satisfies the property $\mathscr{P}_\alpha(\vec{c}_\beta) = \delta_{\alpha,\beta}$, being δ the Kronecker's delta. Defining the inner product of two functions $\mathscr{F}_1(\vec{c})$ and $\mathscr{F}_2(\vec{c})$ as

$$\langle \mathscr{F}_1(\vec{c}), \mathscr{F}_2(\vec{c}) \rangle = \iiint_{-\infty}^{+\infty} \mathscr{F}_1(\vec{c})\mathscr{F}_2(\vec{c})W(\vec{c})\, dc_x dc_y dc_z, \tag{3.3}$$

the orthogonal condition imposes

$$\langle \mathscr{P}_{\alpha_1}(\vec{c}), \mathscr{P}_{\alpha_2}(\vec{c}) \rangle = 0 \qquad \forall \alpha_1 = \alpha_2. \tag{3.4}$$

Applying the polynomial decomposition to the distribution function and weighting with the centred Gaussian probabilistic density function with standard deviation c_s the distribution is decomposed as

$$f(t,\vec{x},\vec{c}) = \frac{1}{(\sqrt{2\pi}c_s)^D}\sum_{\alpha} f_\alpha(t,\vec{x})\mathscr{P}_\alpha(\vec{c})\,\exp\left(-\|\vec{c}\|^2/2c_s^2\right). \tag{3.5}$$

The entire Boltzmann equation 3.1 can be decomposed in analogous way, getting

$$\sum_{\alpha}\left[\left(\partial_t + \vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right) f_\alpha(t,\vec{x}) - \Omega_\alpha(t,\vec{x})\right]\mathscr{P}_\alpha(\vec{c})\,\exp\left(-\|\vec{c}\|^2/2c_s^2\right) = 0. \tag{3.6}$$

Multiplying by other of the polynomials $\mathscr{P}_\beta(\vec{c})$ and integrating in velocity each component is decoupled from the others returning one equation per discrete velocity

$$\left(\partial_t + \vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right) f_\alpha(t,\vec{x}) = \Omega_\alpha(t,\vec{x}). \tag{3.7}$$

To be orthogonal the polynomial roots have to be coincident with the quadrature points. For Gaussian distribution functions the associated orthogonal basis is formed by Hermite polynomials. This requirement is only possible when the stencil is small. Complex schemes with large sets of discrete velocities can not satisfy this condition. Consequently some of the quadrature points are not coincident with the lattice nodes. The orthogonal basis is replaced by polynomials with equidistant roots. With the orthogonality condition not being satisfied the cross product of the polynomials is different to 0

$$\langle\mathscr{P}_\alpha(\vec{c}),\mathscr{P}_\beta(\vec{c})\rangle = \int_{-\infty}^{+\infty}\mathscr{P}_\alpha(c)\mathscr{P}_\beta(c)W(c)\,dc \neq 0. \tag{3.8}$$

After applying the step 3.7 the result is a coupled equation system

$$[M]_{\beta,\alpha}\left(\partial_t + \vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right) f_\alpha(t,\vec{x}) = [M]_{\beta,\alpha}\Omega_\alpha(t,\vec{x}) \tag{3.9}$$

where the weighting matrix $[M]$ is filled instead of being diagonal. If every term in the Boltzmann equation can be directly decomposed following the expression 3.2 the matrix cancels itself. However, if there are some extra terms S

$$\partial_t f + \vec{c}\cdot\vec{\nabla}_{\vec{x}} f = \Omega + S \tag{3.10}$$

and this term can not be decomposed, the resultant expression is

$$\left(\partial_t + \vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right) f_\alpha(t,\vec{x}) - \Omega_\alpha(t,\vec{x}) = [M]^{-1}_{\beta,\alpha}\int_{-\infty}^{+\infty}\mathscr{P}_\beta(c)S(c)\,dc. \tag{3.11}$$

In this case the matrix $[M]$ has to be considered. If it is not the case the procedure can be applied over non-uniform lattices. One example is the work

realised by He and Luo [1]. They decouple the discrete velocity and the lattice. As consequence populations stream to random locations. The interpolation step is introduced in addition to collision and streaming by the researchers. One characteristic of LBM is the relation between the discretisations. The time step Δt, the distance between neighbouring nodes $\Delta\vec{x}$ and the velocity of the particles $\vec{c}$ are related by

$$\Delta\vec{x}_\alpha = \vec{c}_\alpha \Delta t. \tag{3.12}$$

Every particle located in a node can only travel to the exact locations of other nodes during a time step. The selected velocity fixes the geometry of the lattice and vice-versa. A secondary deduction of the previous restriction is the uniformity of the lattice. To keep the velocity set equal in whole domain the distance between the nodes has to be kept too. Considering first order approximation the resultant equation is

$$\begin{aligned}\frac{1}{\Delta t}\left(f_\alpha(t+\Delta t,\vec{x}) - f_\alpha(t,\Delta x)\right)\cdots \\ \cdots + \frac{\vec{c}_\alpha\cdot\Delta\vec{x}_\alpha}{\|\Delta\vec{x}_\alpha\|^2}\left(f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha) - f_\alpha(t+\Delta t,\vec{x})\right) = \Omega_\alpha(t,\vec{x}).\end{aligned} \tag{3.13}$$

$\vec{c}_\alpha$ can be replaced by the restriction 3.12, resulting

$$f_\alpha(t+\Delta t,\vec{x}) - f_\alpha(t,\Delta x) + f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha) - f_\alpha(t+\Delta t,\vec{x}) = \Delta t\,\Omega_\alpha(t,\vec{x}). \tag{3.14}$$

Taking advantage of the linearity, the equation can be decomposed in two whose addition returns the original one

$$f_\alpha(t+\Delta t,\vec{x}\qquad) - f_\alpha(t\qquad,\vec{x}) = \Delta t\,\Omega_\alpha(t,\vec{x}), \tag{3.15a}$$

$$f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha) - f_\alpha(t+\Delta t,\vec{x}) = 0. \tag{3.15b}$$

It represents the division in two different steps of each iteration in time. During the first step the collision operator redistributes the population among the velocities. It solves the evolution of the population in time. During the second step the particles travel to the contiguous nodes following the direction fixed by their velocity. It solves the advective part of the Boltzmann equation. The names given to these steps are **collision step** and **stream step** respectively.

3.2.2 Numerical integration

As it has been seen before, the weighting function for LBM is the centred Gaussian distribution with standard deviation equal to c_s. This values depends on the selected scheme. Every magnitude in LBM is approximated following the expression 3.2. Consequently, the integral of a generic distribution $g(\vec{c})$

can be approximated as

$$\int_{-\infty}^{+\infty} g(\vec{c})\, d\vec{c} \simeq \sum_{\alpha} g_\alpha \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}) W(\vec{c})\, d\vec{c}. \tag{3.16}$$

The integral of the right hand side is always the same. The results are the weighting factors w_α

$$w_\alpha = \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}) W(\vec{c})\, d\vec{c}. \tag{3.17}$$

So the integral value can be calculated fast as a weighted sum

$$\int_{-\infty}^{+\infty} g(\vec{c})\, d\vec{c} \simeq \sum_{\alpha} g_\alpha w_\alpha \tag{3.18}$$

3.2.3 Equilibrium distribution function

In absence of external forces without any gradient in space the ideal monoatomic gas tends to fit the Maxwell distribution. This is a 3D Gaussian distribution centred in the mean velocity whose standard deviation is $\sqrt{k_B T/m}$, being k_B the Boltzmann constant, T the temperature and m the molecular mass. In LBM the standard deviation corresponds to c_s, what is called speed of sound of the lattice. The equilibrium distribution is the Gaussian distribution displaced the macroscopic velocity $\vec{u}$ times the density $\rho(t, \vec{x})$

$$f^{eq}(t, \vec{x}, \vec{c}) = \rho(t, \vec{x}) \exp\left(-\|\vec{c} - \vec{u}\|^2 / 2c_s^2\right) \left(2\pi c_s^2\right)^{-D/2}. \tag{3.19}$$

The polynomial approximation to the equilibrium distribution can be written as

$$f^{eq}(t, \vec{x}, \vec{c}) = \sum_{\alpha} f_\alpha^{eq}(t, \vec{x}) \mathscr{P}_\alpha(\vec{c}) \exp\left(-\|\vec{c}\|^2 / 2c_s^2\right) \left(2\pi c_s^2\right)^{-D/2}. \tag{3.20}$$

Now it is time to compute the discrete values. As usual the previous equation is multiplied by the polynomial $\mathscr{P}_\beta(\vec{c})$ and integrated to get

$$f_\alpha^{eq}(t, \vec{x}) = \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}) f^{eq}(t, \vec{x}, \vec{c})\, d\vec{c}. \tag{3.21}$$

Replacing the equilibrium distribution by its value it is gotten the final expression of the discrete equilibrium distribution

$$f_\alpha^{eq}(t, \vec{x}) = \rho(t, \vec{x}) \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}) \left(2\pi c_s^2\right)^{-D/2} \exp\left(-\frac{\|\vec{c} - \vec{u}\|^2}{2c_s^2}\right) d\vec{c}. \tag{3.22}$$

This equation can be also written in terms of a displaced velocity $\vec{c}' = \vec{c} - \vec{u}$

$$f_\alpha^{eq}(t,\vec{x}) = \rho(t,\vec{x}) \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}' + \vec{u}) \left(2\pi c_s^2\right)^{-D/2} \exp\left(-\frac{\|\vec{c}'\|^2}{2c_s^2}\right) d\vec{c}'. \quad (3.23)$$

The generic polynomial can be written as sum of binomials

$$\mathscr{P}_\alpha(\vec{c}' + \vec{u}) = \sum_n a_n \prod_{d=1}^{D} (c'_d + u_d)^{k_{n,d}}, \quad (3.24)$$

where a_n is the n-th factor and $k_{n,d}$ the n-th exponent for the d-th dimension. Each binomial can be expanded

$$(c'_d + u_d)^{k_{n,d}} = \sum_{m=0}^{k_{n,d}} \binom{k_{n,d}}{m} u_d^m {c'}_d^{k_{n,d}-m} \quad (3.25)$$

where the operand

$$\binom{n}{m} = \frac{n!}{(n-m)!m!} \quad (3.26)$$

is the binomial coefficient. The product of the expansion by the weighting function can be easily integrated. The n-th moment of the d-th independent variable, equivalent to calculate the inner product $\langle {c'}_d^n, 1\rangle$, is known *a priori*

$$M_d^{(n)} = \langle {c'}_d^n, 1\rangle = \int_{-\infty}^{+\infty} {c'}_d^n \left(2\pi c_s^2\right)^{-1/2} \exp\left(-{c'}_d^2/2c_s^2\right) dc'_d. \quad (3.27)$$

It admits algebraic solution

$$M_d^{(n)} = \begin{cases} c_s^n \prod_{m=1}^{n/2} (n-2m+1), & \{n \,|\, (\exists k \in \mathbb{N})[n = 2k]\}; \\ 0, & \{n \,|\, (\exists k \in \mathbb{N})[n = 2k+1]\}. \end{cases} \quad (3.28)$$

It can be observed how the parity of the Gaussian distribution ($W(c) = W(-c)$) makes zero the odd moments. Table 3.1 contains the first even moments of the distribution. The complexity of all this process can be drastically

Moments	$M^{(0)}$	$M^{(2)}$	$M^{(4)}$	$M^{(6)}$	$M^{(8)}$	$M^{(10)}$
Values	1	c_s^2	$3c_s^4$	$15c_s^6$	$105c_s^8$	$945c_s^{10}$

Table 3.1: First six even moments of the centred Gaussian distribution with standard deviation equal to c_s.

reduced if the nodes of the lattice overlap the quadrature points of the weighting function. If this condition is accomplished the integral value of a generic variable can be approximated as

$$\int_{-\infty}^{+\infty} g(x)W(x)\,dx = \sum_{\alpha} w_\alpha g(x_\alpha). \tag{3.29}$$

Each equilibrium component can be reformulated by recovering the expression 3.22 and replacing the exponential by the product of the weighting function and exponential of the deviation

$$f_\alpha^{eq}(t,\vec{x}) = \rho(t,\vec{x}) \int_{-\infty}^{+\infty} \mathscr{P}_\alpha(\vec{c}) \exp\left(\frac{2\vec{c}\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right) \left(2\pi c_s^2\right)^{-D/2} \exp\left(-\frac{\vec{c}\cdot\vec{c}}{2c_s^2}\right) d\vec{c}. \tag{3.30}$$

The integral part can be replaced by the summation of the quadrature

$$f_\alpha^{eq}(t,\vec{x}) = \rho(t,\vec{x}) \sum_{\beta} w_\beta \mathscr{P}_\alpha(\vec{c}_\beta) \exp\left(\frac{2\vec{c}_\beta\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right). \tag{3.31}$$

The α-th polynomial evaluated in the β-th node is $\mathscr{P}_\alpha(\vec{c}_\beta) = \delta_{\alpha\beta}$, where $\delta_{\alpha\beta}$ is the Kronecker's delta. The polynomial takes value 1 when $\alpha = \beta$ and 0 otherwise. The summation is then simplified to a single term

$$f_\alpha^{eq} = \rho w_\alpha \exp\left(\frac{2\vec{c}_\alpha\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right). \tag{3.32}$$

The exponential function is computationally expensive. In practice it is replaced by the truncated Taylor series around $\vec{u} = 0$

$$\exp\left(\frac{2\vec{c}_\alpha\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right) = 1 + \frac{\vec{c}_\alpha\cdot\vec{u}}{c_s^2} - \frac{\vec{u}\cdot\vec{u}}{2c_s^2} + \frac{(\vec{c}_\alpha\cdot\vec{u})^2}{2c_s^4} + \mathscr{O}(\|\vec{u}\|^3). \tag{3.33}$$

Each component of the equilibrium distribution can be calculated as

$$f_\alpha^{eq} = \rho w_\alpha \left[1 + \frac{\vec{c}_\alpha\cdot\vec{u}}{c_s^2} - \frac{\vec{u}\cdot\vec{u}}{2c_s^2} + \frac{(\vec{c}_\alpha\cdot\vec{u})^2}{2c_s^4}\right] \tag{3.34}$$

Another way to approximate the exponential is in terms of the entire exponent

$$\exp\left(\frac{2\vec{c}_\alpha\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{2\vec{c}_\alpha\cdot\vec{u} - \vec{u}\cdot\vec{u}}{2c_s^2}\right)^n. \tag{3.35}$$

It can be also expressed in an recursive way. Being $\mathscr{R}_n(x) = 1/n + x\mathscr{R}_{n+1}(x)$, the exponential is $\exp(x) = 1 + x\mathscr{R}_1(x)$. The calculus starts from the last term, where the recursive series is truncated, with an approximated value $\mathscr{R}_N = 1/N$. The successive terms are computed in descending order applying the recursive rule till calculate $\mathscr{R}_1(x)$ and the exponential. This technique can be used to obtain high order accuracy to admit larger values of $\vec{u}$.

3.2.4 The collision operator

When two population of spherical particles f_i and f_j collides they interchange momentum, and the distribution is altered to f_i' and f_j'. Here the apostrophe means "after collision". The collision operator, considered till now as an abstract entity, admits algebraic solution

$$\Omega_{ij} = \iiint \left[\iint \left(f_i' f_j' - f_i f_j \right) \|\mathbf{c}_i - \mathbf{c}_j\| \sigma_{ij} \, d\omega \right] d\mathbf{c}_j. \tag{3.36}$$

σ_{ij} is the cross section and ω is the solid angle (see figure 3.3). Solving

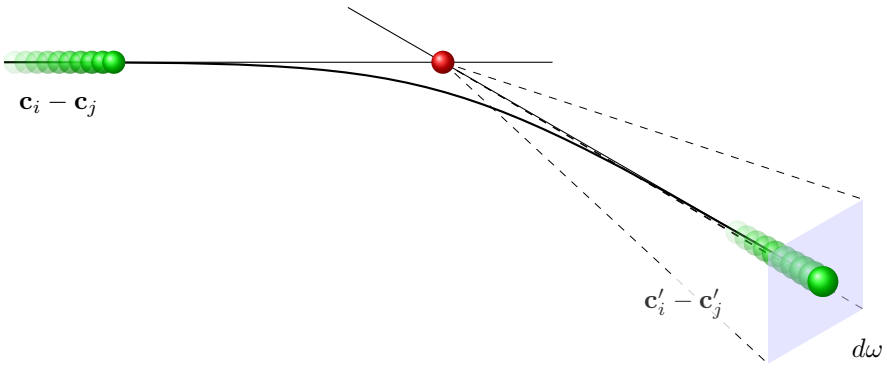

Figure 3.3: Elastic binary collision scheme.

this equation for every node becomes a prohibitive task for several reasons. The post-collision variables are implicit and non-linear, the populations are integrated multiple times and there is a component for every combination of speeds. These are the reasons why in practice the collision operator is modelled instead of being simulated. Several techniques are developed to simplify the calculus. The most important ones are the Single Relaxation Time (SRT) or the so called Bhatnagar-Gross-Krook (BGK), the Two Relaxation Time (TRT) and the Multiple Relaxation Time (MRT). All of them reproduce the asymptotic tendency to the equilibrium distribution induced by the collision operator using relaxation. Several researchers compare the results of the same simulation solved with different collision operators. One example is the work realised by Freitas *et al.* [2]. They solve turbulent channel flow (Couette) for several collision operators. The results are analysed in accuracy and computational speed. The reference values are the results of Direct Numerical Simulation (DNS) simulations.

Turbulent modelling can be included modifying the collision operator. The modification depends on the model. Yu, Girimaji and Luo *et al.* [3] study the decay of isotropic turbulence in cubic domains using both DNS and Large

Eddy Simulation (LES) modelling. For DNS calculation no assumption is required and the collision operator is used without any modification. For LES calculation the collision operator is adapted locally to take into consideration turbulent viscosity.

Bhatnagar-Gross-Krook collision operator

The easiest and fastest model to approximate the collision operator is the BGK model. The equivalence with NS equations was introduced by Qian, d'Humières and Lallemand [4]. It consists on emulating the tendency imposed by Ω over the distribution to tend to equilibrium as a simple relaxation method

$$\Omega = -\frac{1}{\tau}\left(f - f^{eq}\right). \tag{3.37}$$

This approximation can be decomposed following 3.2, obtaining an equivalent expression for each α-th component $\Omega_\alpha = -(f_\alpha - f_\alpha^{eq})/\tau$. The resulting LBM equation is

$$f_\alpha(t + \Delta t, \vec{x} + \Delta\vec{x}_\alpha) - f_\alpha(t, \vec{x}) = -\frac{\Delta t}{\tau}\left[f_\alpha(t, \vec{x}) - f_\alpha(t, \vec{x})\right]. \tag{3.38}$$

Two Relaxation Time collision operator

TRT collision model improves the robustness of BGK. The distribution is decomposed in the symmetric f_α^s and anti-symmetric f_α^a parts

$$f_\alpha = f_\alpha^s + f_\alpha^a. \tag{3.39}$$

The way to calculate each component is

$$f_\alpha^s = (f_\alpha + f_{\bar{\alpha}})/2, \qquad f_\alpha^a = (f_\alpha - f_{\bar{\alpha}})/2; \tag{3.40}$$

where $\bar{a}$ means relative to the opposite α-th discrete velocity $\vec{c}_{\bar{\alpha}} = -\vec{c}_\alpha$. Then two different interrelated relaxation parameters are applied to each population.

$$\Omega_\alpha = -\frac{1}{\tau_s}\left(f_\alpha^s - f_\alpha^{s,eq}\right) - \frac{1}{\tau_a}\left(f_\alpha^a - f_\alpha^{a,eq}\right). \tag{3.41}$$

The parameter τ_s is the equivalence of τ for the BGK model. τ_s and τ_a are related by a magic number χ

$$\chi = \left(\frac{1}{\tau_s} + \frac{1}{2}\right)\left(\frac{1}{\tau_a} + \frac{1}{2}\right). \tag{3.42}$$

This number receives different values attending to the objective criteria. Walls located at the middle distance between two contiguous nodes are correctly calculated for $\chi = 3/16$. Values of $\chi = 1/4$ returns the strongest robustness.

Multiple Relaxation Time collision operator

One further step is to describe the relaxation process by means of several relaxation parameters. What MRT does is to relax the integral moments in a different way instead of applying directly the relaxation on the distribution. The process is divided in three steps. Firstly the integral moments of interest $\boldsymbol{m}$ are calculated as a weighted sum of the discrete components of the distribution $\boldsymbol{m} = [M]\boldsymbol{f}$. Mathematically this operation is the product of a matrix $[M]$ and the vector which contains the distribution $\boldsymbol{f}$. The resultant moments are relaxed independently, each one with its own relaxing factor λ. Once the moments have been relaxed the new distribution is recovered multiplying by $[M]^{-1}$

$$\boldsymbol{\Omega} = [M]^{-1}[\Lambda]([M]\boldsymbol{f} - \boldsymbol{m}^{eq}). \tag{3.43}$$

The relaxing parameters are contained in the diagonal matrix $[\Lambda]$. In a similar way to the TRT model, each factor is selected accordingly to an objective criterion such as accuracy or robustness. Because the extra tuning parameters the MRT can be also adapted to multiphysics.

3.2.5 Chapman-Enskog expansion

Fluid mechanics is governed by the NS equation. The LBM approximates this behaviour. It is needed to find the equivalence between both formulations. The followed process to find this relation is the Chapman-Enskog expansion.

The NS equations for mass and moment conservation are

$$\partial_t \rho + \vec{\nabla} \cdot (\rho \vec{u}) = 0 \qquad \text{and} \tag{3.44a}$$

$$\partial_t (\rho \vec{u}) + \vec{\nabla} \cdot (\rho \vec{u} \otimes \vec{u}) = -\vec{\nabla} P + \mu \vec{\nabla} \cdot \left[\vec{\nabla} \otimes \vec{u} + (\vec{\nabla} \otimes \vec{u})^T \right] \tag{3.44b}$$

respectively. Here the gradient is denoted by $\vec{\nabla}\otimes$ instead of simply $\vec{\nabla}$ to be easily distinguished from the divergence $\vec{\nabla}\cdot$. The operator $\otimes$ is the external product, P is the pressure, μ the dynamic viscosity, ρ the density and $\vec{u}$ the velocity.

The equivalence depends on the numerical scheme and the collision model. From now in advance all the formulation will be developed for the BGK collision model for the sake of simplicity. It is the simplest model and reaches maximum performance. However there is a penalty in accuracy and stability for high Reynolds numbers. It is not the case of the simulations of this thesis, where the Reynolds number is around 1. This regime is completely laminar and it is close to be Stokes regime. BGK collision model returns satisfactory results in this context. The process for other collision models is analogous.

The discrete Boltzmann equation without external forces simplified with the BGK model is

$$f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)-f_\alpha(t,\vec{x})=-\frac{\Delta t}{\tau}\left(f_\alpha(t,\vec{x}_\alpha)-f_\alpha^{eq}(t,\vec{x}_\alpha)\right) \tag{3.45}$$

The displaced term can be expanded around $f_\alpha(t,\vec{x})$ in Taylor series as

$$f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)=\sum_{n=0}^{\infty}\frac{1}{n!}\left(\Delta t\partial_t+\Delta\vec{x}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)^n f_\alpha(t,\vec{x}). \tag{3.46}$$

where the n-th power of the gradient term is

$$\left(\Delta\vec{x}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)^n=\Delta\vec{x}_\alpha{}^{\otimes n}\cdot\vec{\nabla}_{\vec{x}}^{\otimes n} \tag{3.47}$$

Expanding the series and truncating in second order the expression becomes

$$\begin{aligned}f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)=\Bigg(1+\Delta t\partial_t+\Delta\vec{x}_\alpha\cdot\vec{\nabla}_{\vec{x}}+\frac{1}{2}\Delta t^2\partial_t^2\cdots\\ \cdots+\Delta t\Delta\vec{x}_\alpha\cdot\vec{\nabla}_{\vec{x}}+\frac{1}{2}\Delta\vec{x}_\alpha{}^{\otimes 2}\cdot\vec{\nabla}_{\vec{x}}^{\otimes 2}\Bigg)f_\alpha(t,\vec{x})+\mathscr{O}(\Delta t^2,\|\Delta\vec{x}_\alpha\|^2).\end{aligned} \tag{3.48}$$

From restriction 3.12 the space increment can be replaced as $\Delta\vec{x}_\alpha=\Delta t\vec{c}_\alpha$. The previous equation becomes

$$\begin{aligned}f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)=\Bigg(1+\Delta t\partial_t+\Delta t\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}+\frac{1}{2}\Delta t^2\partial_t^2\cdots\\ \cdots+\Delta t^2\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}+\frac{1}{2}\Delta t^2\vec{c}_\alpha^{\otimes 2}\cdot\vec{\nabla}_{\vec{x}}^{\otimes 2}\Bigg)f_\alpha(t,\vec{x})+\mathscr{O}(\Delta t^2).\end{aligned} \tag{3.49}$$

The same equation can be written as a function of the operator $\partial_t+\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}$ as

$$\begin{aligned}f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)\simeq\Bigg[1+\Delta t\left(\partial_t+\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)+\cdots\\ \cdots\frac{\Delta t^2}{2}\left(\partial_t+\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)^2\Bigg]f_\alpha(t,\vec{x})+\mathscr{O}(\Delta t^2).\end{aligned} \tag{3.50}$$

The value of $f_\alpha(t+\Delta t,\vec{x}+\Delta\vec{x}_\alpha)$ can be replaced in expression 3.45 returning

$$\left[\left(\partial_t+\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)+\frac{\Delta t}{2}\left(\partial_t+\vec{c}_\alpha\cdot\vec{\nabla}_{\vec{x}}\right)^2\right]f_\alpha(t,\vec{x})=-\frac{1}{\tau}\left(f_\alpha(t,\vec{x})-f_\alpha^{eq}(t,\vec{x})\right) \tag{3.51}$$

The distribution, as well as the derivative operators, can be decomposed as the addition of components of different order of magnitude

$$f=f^{(0)}+\varepsilon f^{(1)}+\varepsilon^2 f^{(2)}, \tag{3.52a}$$

$$\partial_t=\varepsilon\partial_t^{(0)}+\varepsilon^2\partial_t^{(1)}, \tag{3.52b}$$

$$\vec{\nabla}_{\vec{x}}=\varepsilon\vec{\nabla}_{\vec{x}}^{(0)}. \tag{3.52c}$$

The previous expansions in order of magnitude are included in 3.51. As result, one equation is returned for each order of magnitude:

$$\varepsilon^0: \quad f_\alpha^{(0)} - f_\alpha^{eq} = 0 \tag{3.53a}$$

$$\varepsilon^1: \quad -\frac{1}{\tau} f_\alpha^{(1)} = \left(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}\right) f_\alpha^{(0)} \tag{3.53b}$$

$$\varepsilon^2: \quad -\frac{1}{\tau} f_\alpha^{(2)} = \left(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}\right) f_\alpha^{(1)} + \cdots \tag{3.53c}$$

$$\cdots \frac{\Delta t}{2}\left(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}\right)^2 f_\alpha^{(0)} + \partial_t^{(1)} f_\alpha^{(0)}$$

The second equation can be introduced in the third replacing $(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}) f_\alpha^{(0)}$ by $-f_\alpha^{(1)}/\tau$

$$\varepsilon^0: \quad f_\alpha^{(0)} - f_\alpha^{eq} = 0 \tag{3.54a}$$

$$\varepsilon^1: \quad -\frac{1}{\tau} f_\alpha^{(1)} = \left(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}\right) f_\alpha^{(0)} \tag{3.54b}$$

$$\varepsilon^2: \quad -\frac{1}{\tau} f_\alpha^{(2)} = \left(1 - \frac{\Delta t}{2\tau}\right)\left(\partial_t^{(0)} + \vec{c}_\alpha \cdot \vec{\nabla}_{\vec{x}}^{(0)}\right) f_\alpha^{(1)} + \partial_t^{(1)} f_\alpha^{(0)} \tag{3.54c}$$

From the first expression it is concluded that the distribution of 0 order is the equilibrium distribution. The higher orders will be used to fit the relation with NS. The first moments of the order 0 distribution are, by definition,

$$\rho = \sum_\alpha f_\alpha^{(0)}, \qquad \rho\vec{u} = \sum_\alpha \vec{c}_\alpha f_\alpha^{(0)}, \qquad \Pi^{(0)} = \sum_\alpha \vec{c}_\alpha^{\otimes 2} f_\alpha^{(0)},$$
$$\Xi^{(0)} = \sum_\alpha \vec{c}_\alpha^{\otimes 3} f_\alpha^{(0)}, \qquad 0 = \sum_\alpha f_\alpha^{(n)} \quad \forall\{n \in \mathbb{N} | n > 0\}$$
$$\text{and} \qquad 0 = \sum_\alpha \vec{c}_\alpha f_\alpha^{(n)} \quad \forall\{n \in \mathbb{N} | n > 0\}. \tag{3.55}$$

Here $\Pi^{(0)}$ and $\Xi^{(0)}$ are tensors of second and third order respectively. The n-th moment of a discrete variable g_α is a tensor of order n calculated as $M^{(n)}(g_\alpha) = \sum_\alpha \vec{c}^{\otimes n} \otimes g_\alpha$. This definition can be used to calculate the zeroth, first and second moments of expression 3.54b

$$0 = \partial_t^{(0)}\rho \quad + \vec{\nabla}_{\vec{x}}^{(0)} \cdot (\rho\vec{u}), \tag{3.56a}$$

$$0 = \partial_t^{(0)}(\rho\vec{u}) + \vec{\nabla}_{\vec{x}}^{(0)} \cdot \Pi^{(0)}, \tag{3.56b}$$

$$-\frac{1}{\tau}\Pi^{(1)} = \partial_t^{(0)}\Pi^{(0)} + \vec{\nabla}_{\vec{x}}^{(0)} \cdot \Xi^{(0)}. \tag{3.56c}$$

Similar procedure can be performed with expression 3.54c returning

$$0 = \partial_t^{(1)}\rho, \tag{3.57a}$$

$$0 = \left(1 - \frac{\Delta t}{2\tau}\right)\vec{\nabla}_{\vec{x}}^{(0)} \cdot \Pi^{(1)} + \partial_t^{(1)}\rho\vec{u}. \tag{3.57b}$$

The macroscopic integral equation can be recovered by the addition of the moments

$$0 = \left(\varepsilon\partial_t^{(0)} + \varepsilon^2\partial_t^{(1)}\right)\rho \quad + \varepsilon\vec{\nabla}_{\vec{x}}^{(0)} \cdot (\rho\vec{u}), \tag{3.58a}$$

$$0 = \left(\varepsilon\partial_t^{(0)} + \varepsilon^2\partial_t^{(1)}\right)(\rho\vec{u}) + \varepsilon\vec{\nabla}_{\vec{x}}^{(0)} \cdot \left[\Pi^{(0)} + \varepsilon\left(1 - \Delta t/2\tau\right)\Pi^{(1)}\right]. \tag{3.58b}$$

The original derivatives can be recovered

$$0 = \partial_t\rho \quad + \vec{\nabla}_{\vec{x}} \cdot (\rho\vec{u}), \tag{3.59a}$$

$$0 = \partial_t(\rho\vec{u}) + \vec{\nabla}_{\vec{x}} \cdot \left[\Pi^{(0)} + \varepsilon\left(1 - \Delta t/2\tau\right)\Pi^{(1)}\right]. \tag{3.59b}$$

$\Pi^{(0)}$ and $\Xi^{(0)}$ are defined as the moments of the equilibrium distribution 3.19

$$M^{(n),eq} = \int_{-\infty}^{+\infty} \vec{c}^{\otimes n} f^{eq}(\vec{c})\, d\vec{c}. \tag{3.60}$$

The integral returns the values

$$\Pi_{ij}^{(0)} = M^{(2),eq} = \rho\theta\delta_{ij} + \rho u_i u_j \tag{3.61a}$$

$$\Xi_{ijk}^{(0)} = M^{(3),eq} = \rho\theta\left(u_i\delta_{jk} + u_j\delta_{ik} + u_k\delta_{ij}\right) + \rho u_i u_j u_k \tag{3.61b}$$

where $\theta = c_s^2$ and δ is the Kronecker's delta. These values can be replaced in 3.56c getting

$$-\frac{1}{\tau}\Pi_{ij}^{(1)} = \partial_t^{(0)}\left(\rho\theta\delta_{ij} + \rho u_i u_j\right) + \partial_{x_k}^{(0)}\left(\rho\theta\left(u_i\delta_{jk} + u_j\delta_{ik} + u_k\delta_{ij}\right) + \rho u_i u_j u_k\right). \tag{3.62}$$

Assuming some hypothesis the previous equation can be reduced:

- $\theta = c_s^2$ is constant (isothermal flow).

  $$\partial_t\rho\theta + \vec{\nabla}_{\vec{x}} \cdot \rho\theta\vec{u} = \theta\left(\partial_t\rho + \vec{\nabla}_{\vec{x}} \cdot \rho\vec{u}\right) = 0 \tag{3.63}$$

  The continuity equation makes 0 the elements of the trace $\partial_t\rho\theta\delta_{ij} + \partial_{x_k}\rho\theta u_k\delta_{ij} = 0$.

- The velocity magnitude is very small. This is equivalent to assume constant velocity cross product $u_i u_j$.

  $$\partial_t(\rho u_i u_j) + \partial_{x_k}\rho u_i u_j u_k = u_i u_j\left(\partial_t\rho + \vec{\nabla} \cdot (\rho\vec{u})\right) = 0 \tag{3.64}$$

  Second and higher terms are negligible, $\partial_t\rho u_i u_j + \partial_{x_k}\rho u_i u_j u_k \simeq 0$.

The equation is then reduced to

$$-\frac{1}{\tau}\Pi_{ij}^{(1)} = \theta(\partial_{x_i}\rho u_j + \partial_{x_j}\rho u_i). \tag{3.65}$$

Written as tensor products the equation is

$$-\frac{1}{\tau}\Pi^{(1)} = \theta\left(\vec{\nabla}\otimes\rho\vec{u} + (\vec{\nabla}\otimes\rho\vec{u})^T\right). \tag{3.66}$$

Now the tensors $\Pi^{(0)}$ and $\Pi^{(1)}$ can be replaced in the momentum equations to obtain

$$0 = \partial_t\rho + \vec{\nabla}\cdot(\rho\vec{u}), \tag{3.67a}$$

$$0 = \partial_t(\rho\vec{u}) + \vec{\nabla}\cdot\left[\rho\theta[I] + \rho\vec{u}\otimes\vec{u} - \varepsilon\theta\left(\tau - \frac{\Delta t}{2}\right)\left(\vec{\nabla}\otimes\rho\vec{u} + (\vec{\nabla}\otimes\rho\vec{u})^T\right)\right]. \tag{3.67b}$$

This equation takes the same shape than NS. However the magnitudes are not directly equivalent. NS works with physical units. LBM works with lattice units. For a direct comparison it is needed to scale NS to lattice units.

3.2.6 Scaling Navier-Stokes to lattice units

The LBM works with lattice units. It improves the performance and simplifies the formulation. In this section the physical units are denoted by the superscript ^{Ph}. There are three basic scaling factors to convert the physical units into lattice units.

- **Length conversion.** Two contiguous nodes are separated by one length lattice unit. The conversion factor is then directly

  $$x^{Ph} = x\Delta x \tag{3.68}$$

  where Δx is the lattice size.

- **Velocity conversion.** The time step is one time lattice unit long. During this time the characteristic distribution covers one length lattice unit. In other words, the characteristic speed of the distribution is one length lattice unit per time lattice unit. Accordingly with the LBM hypothesis, the characteristic macroscopic speed U_0 has to be much lower than the speed of the distribution. Usually a factor K is selected in such a way that K velocity lattice units are equivalent to U_0 physical units

  $$u^{Ph} = U_0 u/K. \tag{3.69}$$

  K usually takes values around 0.01.

- **Time conversion.** During a time step the distribution with velocity U_0/K physical units covers the distance Δx. The time step is one time lattice unit long, consequently

$$t^{Ph} = \frac{K\Delta x}{U_0} t. \tag{3.70}$$

- **Density and mass conversion.** The surrounding volume around a node contains one mass lattice unit. In physical units this value is $\rho_0^{Ph}\Delta x^3$. Then the relation between the mass lattice unit and its physical counterpart is

$$m^{Ph} = \rho_0^{Ph}\Delta x^3 m. \tag{3.71}$$

  The relation between densities is immediate

$$\rho^{Ph} = \rho_0^{Ph}\rho. \tag{3.72}$$

The physical magnitudes expressed in lattice units can be replaced in the NS equation 3.44. After dividing the resultant expression by $\rho_0 U_0^2/K^2\Delta x$ it is obtained

$$\partial_t \rho + \vec{\nabla}\cdot(\rho\vec{u}) = 0 \qquad \text{and} \tag{3.73a}$$

$$\partial_t(\rho\vec{u}) + \vec{\nabla}\cdot(\rho\vec{u}\otimes\vec{u}) = -\frac{K^2}{\rho_0 U_0^2}\vec{\nabla}P^{Ph} + \frac{1}{Re_L}\frac{L}{\Delta x}K\vec{\nabla}\cdot\left[\vec{\nabla}\otimes\vec{u} + (\vec{\nabla}\otimes\vec{u})^T\right], \tag{3.73b}$$

where the Reynolds number Re_L is

$$Re_L = \frac{\rho_0 U_0 L}{\mu}. \tag{3.74}$$

The expression 3.73 is directly comparable with the results obtained from the Chapman-Enskog expansion 3.67.

$$\begin{array}{|c|}\hline \partial_t(\rho\vec{u}) \\ \partial_t(\rho\vec{u}) \\ \text{Variation} \\ \hline \end{array} + \begin{array}{|c|}\hline \vec{\nabla}\cdot(\rho\vec{u}\otimes\vec{u}) \\ \vec{\nabla}\cdot(\rho\vec{u}\otimes\vec{u}) \\ \text{Convection} \\ \hline \end{array} = \begin{array}{|c|}\hline -\frac{K^2}{\rho_0 U_0^2}\vec{\nabla}P \\ -\vec{\nabla}\cdot(\rho\theta[I]) \\ \text{Pressure} \\ \hline \end{array} + \begin{array}{|c|}\hline \frac{1}{Re_L}\frac{L}{\Delta x}K\vec{\nabla}\cdot\left[\left(\vec{\nabla}\otimes\vec{u} + (\vec{\nabla}\otimes\vec{u})^T\right)\right] \\ \vec{\nabla}\cdot\left[\varepsilon\theta\left(\tau - \frac{\Delta t}{2}\right)\left(\vec{\nabla}\otimes\rho\vec{u} + (\vec{\nabla}\otimes\rho\vec{u})^T\right)\right] \\ \text{Stress} \\ \hline \end{array} \tag{3.75}$$

The derivative in time and the advection terms are directly equivalent. The rest of the terms have to be compared, forcing the factors to be equal.

- **Pressure conversion.** The pressure conversion can be calculated by direct comparison of the "gradient of scalar" term between NS equation and Chapman-Enskog expansion

$$\frac{K^2}{\rho_0 U_0^2}\vec{\nabla}P^{Ph} = \vec{\nabla}\rho c_s^2. \tag{3.76}$$

Because all the factors are constant, the gauge pressure is

$$P^{Ph} = (\rho - 1)c_s^2 \frac{\rho_0 U_0^2}{K^2}. \tag{3.77}$$

- **Viscosity conversion.** The divergence of the sum of gradients term can be also directly compared

$$\frac{1}{Re_L}\frac{L}{\Delta x}K\vec{\nabla}\cdot\left[\vec{\nabla}\otimes\vec{u}+(\vec{\nabla}\otimes\vec{u})^T\right]\cdots$$
$$\cdots=\theta\left(\tau-\frac{\Delta t}{2}\right)\vec{\nabla}\cdot\left[\vec{\nabla}\otimes\rho\vec{u}+(\vec{\nabla}\otimes\rho\vec{u})^T\right] \tag{3.78}$$

  It can be noticed how the density is implicit in the right hand side. Consequently these terms are no directly comparable if the density gradient is not negligible. Considering the density as constant and equal to 1 in lattice units the equivalence is

$$\mu = \frac{c_s^2 \rho_0 U_0 \Delta x}{K}\left(\tau - \frac{\Delta t}{2}\right). \tag{3.79}$$

  For dimensionless problems characterised by the Reynolds number the equivalence is

$$\frac{1}{Re_L}\frac{L}{\Delta x}K = c_s^2\left(\tau - \frac{\Delta t}{2}\right). \tag{3.80}$$

3.2.7 Summary of hypothesis

Several hypothesis have been considered in this section. There are several reasons to assume them. Compatibility with NS equations, improvement of the performance, robustness, discretisation... The most important ones are summarised here. Every simulation has to accomplish these conditions to be correctly solved. The following hypothesis have to be considered before calculating with LBM:

- Characteristic velocity much lower than distribution velocity $U_0 \ll \vec{c}_\alpha$. This hypothesis has been assumed along the whole process.
  - Taylor series expansion of the equilibrium distribution around $\vec{u} = 0$. The equilibrium distribution has been decomposed as the product of equilibrium for velocity zero (weighting function) times the deviation. The deviation has been Taylored to reduce the computational cost. Large velocities imply error in the equilibrium distribution.

  - Negligible second moment variation. The tensor terms $\partial_t(\rho u_i u_j) + \partial_{x_k}(\rho u_i u_j u_k)$ appears during the Chapman-Enskog expansion. Both of them have been considered negligible in comparison with other terms where the velocity appears only once.

- Isotherm flow. The value of c_s has been considered constant. The terms $\partial_t \rho\theta + \vec{\nabla} \cdot (\rho\theta\vec{u})$ are zero with this assumption. However variations of c_s can be considered. The deviation respect to the reference weighting function has to include these effects.

- Quasi-incompressible. The variations in density has been considered small. This assumption is linked with the low velocity hypothesis (large variations of velocity induces large variations of density). This hypothesis is needed to force compatibility between NS and Boltzmann equation. More complex models can simulate compressible effects. Xu *et al.* [5] summarise in their review some of these methods.

3.2.8 Lattice Boltzmann *vs.* Classical CFD

Using LBM instead of classical CFD techniques like Finite Volume Method (FVM) is advantageous depending on the context. There are several factors to take into consideration before deciding which method is the most appropriate. The decision depends on the characteristics of each technique. In the following lines LBM is briefly compared with Finite Volume Method, the most used in fluid dynamics.

- **Linearity**: FVM solves the discrete NS equations. These equations have second order terms in velocity. On the contrary, LBM solves Boltzmann equation, which is completely linear in the implicit component.

- **Correction of pressure loop**: pressure and velocity can not be computed at once with NS formulation. An internal loop of iterations is needed to correct the pressure and achieve mass conservation. This step is not needed with LBM.

- **Boundary conditions**: the implementation of BCs for LBM is much simpler than for FVM. In addition LBM BCs are usually under-restricted problems.

- **Immersed geometry**: the discretisation in LBM has not to be adjusted to the walls. Consequently complex geometries can be meshed easier and moving walls does not force to remesh the domain each iteration.

- **Simpler implementation**: the mathematics of LBM are much easier than NS. The development is faster and robuster.

- **Locality**: the formulation of LBM is explicit. FVM needs to be implicit to have long enough time steps with accurate results. Implicit formulation requires inversion of large matrices. The operation is global. On the contrary, LBM is local. Each discrete element can be calculated independently of the rest. As consequence the communication between processors in LBM is much lower than in FVM. It means that the scalability of LBM methods is close to be ideal [6].

- **More memory consumption**: LBM has more registers than FVM. Consequently memory consumption is higher in LBM than in FVM.

- **Unsteady solver**: LBM is intrinsically unsteady.

The previous points can be considered to argue when to select the LBM as the best option. There are several factors to take into account: computer architecture, size of the problem and complexity of the geometry. LBM should be used under these assumptions:

- **Expensive calculations**: scalability is one of the most important advantages of LBM methods. When the number of discrete elements becomes very large it is recommendable to use LBM instead of FVM. The reasons are better scalability (usually large simulations runs in several cores) and locality of the algorithm. Inversion of matrices, required by FVM, becomes more expensive when the number of discrete elements increases.

- **Complex geometry**: complex geometries like porous media are very difficult to be correctly meshed. This is the reason why LBM becomes a better option. Something similar happens with moving walls. The mesh in FVM has to be refreshed for each time step. Remeshing is not required in LBM.

- **Computers with accelerators**: LBM algebra is very easy to vectorise. Consequently it is the best candidate to be implemented for accelerators as GPUs.

Several researchers are trying to improve the advantages and reduced the weaknesses of LBM. The most important disadvantage is memory consumption. Several researches focus their attention in reducing the demand of memory. Argentini, Bakker and Lowe [7] propose an algorithm for structured lattices which can save up to 78% of the required memory.

3.3 Boundary Conditions

When the surrounding area around a lattice node is not part of the domain the incoming distributions from those directions do not exist. The stream step can not be completed. The BCs built the nonexistent part accordingly to the existent portion and the restrictions at the boundaries. Some kinds of BCs also recompute the entire distribution.

Unlike classical CFD techniques, the BCs of LBM are usually under-restricted. It means that the conditions can be achieved by multiple distributions. The article of Latt *et al.* [8] probes this. They compare several formulations for the same BCs. It is analysed the performance and the accuracy. Therefore extra constrains are required to restrict the problem. The election of these constrains depends on the desired functionality. Robustness, accuracy or computational cost are typical criteria. This is expressed by means of an objective function. The objective functions are either extra restrictions or optimisation functions. The same objective function applied for different restrictions creates families of BCs. The most important sets of BCs are the Inamuro BCs [9] and the Zou & He BCs [10]. The bounce-back BC, only applicable for stopped walls, is also widely used due to its simplicity. All of them are local and they only create or modify the unknown part of the distribution. These three families and a fourth one developed in this thesis are explained in the following sections. An example makes the explanation easier. It is shown in figure 3.4. It consists on a vertical wall located at the right of the central node, at half lattice spacing away, in a D2Q9 scheme. The procedure followed for this particular case can be extrapolated to any other generic scheme.

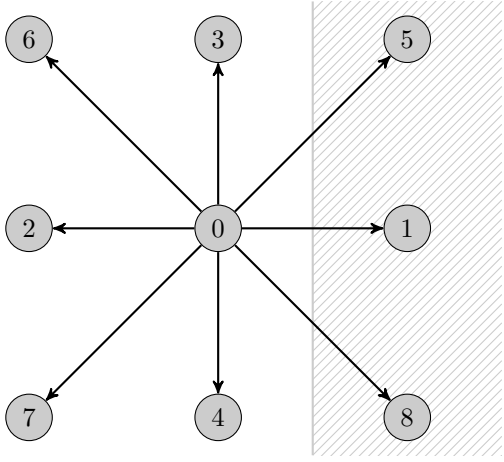

Figure 3.4: Boundary condition scheme.

More sophisticated ways to restrict the BCs exist. The explanation about them is not included in this document but a mention is needed. They usually improve robustness or accuracy paying a penalty in performance. Interpolated and extrapolated distributions are generally more accurate for smooth limiting surfaces. This is the case of Guo BCs [11]. Simplest BCs return exact solutions if the wall is located at the middle distance between neighbouring nodes. Guo *et al.* force the restriction in the exact location by interpolation. This interpolation locates the wall with first order accuracy. Higher orders can be achieved using finite differences. The extrapolated values contains information from several nodes. It is taken into account not only the values of the surrounding nodes but also the first derivatives. The resulting BC is more stable and accurate. The negative counterpart is the lost of locality. Skordos [12] developed BCs in analogous way. He uses the gradient to extrapolate values outside the domain.

In spite of more accurate results and better stability for smooth walls, high order interpolation techniques are not recommended for sharp surfaces. High curvature induces aliasing and other negative effects. Consequently the numerical error of high order techniques is greater for sharp walls in comparison with first order interpolation or the most basic methods.

3.3.1 Fixed wall: bounce-back

The bounce-back BC is the easiest and fastest way to simulate walls. For stopped boundaries, the bounce-back BC consists on inverting the distribution which impinges on it $f_\alpha^{\text{out}} = f_{\bar{\alpha}}$. The subscript $_{\bar{\alpha}}$ means "related to the opposite α-th component, whose speed is $\vec{c}_{\bar{\alpha}} = -\vec{c}_\alpha$". The mean value between the mirrored and the streamed population is directly 0, reason why the zero speed condition is directly accomplished. Because the outgoing population is the same than the incoming one, but with opposite speed, the conservation of mass is also achieved.

With respect to the implementation, there are two variants. One of them, the half-way bounce-back, reintroduces the population into the domain during the same time step. The outgoing distributions are bounced-back into the source node but with opposite direction. Only some of the components are affected by the BC. The second strategy consists on transferring the impinging components into a node located outside the domain. Once it has been transferred, whole the population is inverted and streamed again. This method is called full-way bounce-back. With this strategy it is not necessary to check when a direction cuts the wall. However, if the outer nodes are nodes of the lattice, every full-way bounce-back BC needs to be at least one cell thick.

Both bounce-back BCs have exact solution when the wall is located at the

middle distance between nodes. Unfortunately this is not the usual case. The exact bounced distribution usually finishes in a non-valid location out of the lattice. Some advance BCs redistribute the bounced population between the neighbouring nodes. It makes the BC more accurate for smooth surfaces. The negative counterpart is its penalty in performance. Extra operations are required to redistribute the populations. For the sake of simplicity and other reasons related with the geometry (see section 3.3.5 for more details), pure bounce-back BC has been considered good enough for the simulations of interest.

With respect to the formulation purposed here, the distribution of reference is the full-bounced distribution $f_\alpha^{\text{ref}} = f_{\bar{\alpha}}$. Its density is conserved $\rho^{\text{out}} - \rho^{\text{ref}} = 0$ as well as the moment $(\rho\vec{u})^{\text{out}} - (\rho\vec{u})^{\text{ref}} = 0$. It can be seen how the original bounce-back expression is returned $f_\alpha^{\text{out}} = f_\alpha^{\text{ref}} = f_{\bar{\alpha}}$. After calculating f_α^{out} only the components which impinge the wall are streamed to the fluid node.

3.3.2 Non-slip wall. Inamuro boundary condition.

Inamuro, Yoshino and Ogino [9] proposed in 1995 a new BC for moving walls. The extra condition required to restrict the problem is equilibrium. Every incoming population is completely relaxed.

The Inamuro BC is based on its counterpart in the kinetic theory. In molecular dynamics, when a population strikes the wall it is reflected diffusely following the Maxwellian distribution (equilibrium). The reflected distribution moves at the velocity of the wall. However the resultant population does not. In molecular dynamics there is a discrepancy between the velocity of the wall and the surrounding gas. It happens when the distance from the wall is comparable to the mean free path. This difference in velocity, tangent to the wall, is called slip velocity. It contradicts the restriction imposed by the BC. In some sense calculating non-slip BCs by means of equilibrium is a paradox.

Inamuro *et al.* reformulate the reflection to compensate the slip velocity by means of a mathematical artifice. This undesired magnitude, unknown at the beginning, is added to the wall velocity before calculating equilibrium. The normal velocity does not change. To make the explanation more intuitive from now in advance the mathematical development is specified for the reference example shown at figure 3.4. The unknown components are f_2, f_6 and f_7. All of them are in equilibrium. They can be written in terms of the velocity of the

wall $\vec{u}_w$, the slip velocity $\vec{u}'$ and the corrected density ρ' as

$$f_2^{\text{eq}} = \rho' \frac{1}{9}\left[1 + 3u_w + \frac{9}{2}u_w^2 - \frac{3}{2}\left[u_w^2 + (v_w + v')^2\right]\right], \tag{3.81}$$

$$f_6^{\text{eq}} = \rho' \frac{1}{36}\left[1 + 3\left(-u_w + v_w + v'\right)\cdots \right. \tag{3.82}$$
$$\left. \cdots + \frac{9}{2}\left(-u_w + v_w + v'\right)^2 - \frac{3}{2}\left[u_w^2 + (v_w + v')^2\right]\right],$$

$$f_7^{\text{eq}} = \rho' \frac{1}{36}\left[1 + 3\left(-u_w - v_w - v'\right)\cdots \right. \tag{3.83}$$
$$\left. \cdots + \frac{9}{2}\left(-u_w - v_w - v'\right)^2 - \frac{3}{2}\left[u_w^2 + (v_w + v')^2\right]\right].$$

The variables ρ' and v' are the unknowns to be found. A peculiarity of v' is that the equation becomes non-linear when the walls are not parallel to the lattice. This is the reason why it is not recommended to use this BC to solve oblique limits. The solutions of ρ' and v' for the case of reference (the second order term of v' is cancelled) are

$$\rho_w = \frac{1}{1 - u_w}\left[f_0 + f_3 + f_4 + 2\left(f_1 + f_5 + f_8\right)\right], \tag{3.84a}$$

$$\rho' = 6\frac{\rho_w u_w + \left(f_1 + f_5 + f_8\right)}{1 + 3u_w + 3u_w^2}, \tag{3.84b}$$

$$v' = \frac{1}{1 + 3u_w}\left[6\frac{\rho_w u_w - \left(f_3 - f_4 + f_5 - f_8\right)}{\rho'} - v_w - 3u_w v_w\right]. \tag{3.84c}$$

After computing the slip velocity and the density they are replaced into the previous expression to obtain the unknown distributions.

3.3.3 Zoe & He boundary conditions

Zoe & He developed (1995) a complete set of BCs based on the reflection of the non-equilibrium part. For the case of reference of figure 3.4, where f_2, f_6 and f_7 are unknowns, the generic equations for all of them are

$$f_2 + f_6 + f_7 = \rho^{\text{obj}} - \left(f_0 + f_1 + f_3 + f_4 + f_5 + f_8\right) \tag{3.85a}$$

$$f_2 + f_6 + f_7 = -\left(\rho u\right)^{\text{obj}} + f_1 + f_5 + f_8 \tag{3.85b}$$

$$f_6 - f_7 = \left(\rho v\right)^{\text{obj}} + f_3 - f_4 + f_5 - f_8 \tag{3.85c}$$

It can be observed how the first and second equations are redundant. Zoe & He force bounce-back of the non-equilibrium part of the component normal to the boundary to restrict the problem $f_2 - f_2^{\text{eq}} = f_1 - f_1^{\text{eq}}$. If density ρ^{obj} and velocity $\vec{u}^{\,\text{obj}}$ are known the equilibrium distribution is also known. The

remaining unknown is easily calculated as $f_2 = f_1 + f_2^{\mathrm{eq}} - f_1^{\mathrm{eq}}$. The equilibrium values are replaced by the corresponding expanded expressions in terms of the density and velocity (see equation 3.34). After solving the system the solution is

$$f_2 = f_1 - \frac{2}{3}(\rho u)^{\mathrm{obj}}, \tag{3.86a}$$

$$f_6 = f_5 + \frac{1}{2}(f_3 - f_4) - \frac{1}{6}(\rho u)^{\mathrm{obj}} + \frac{1}{2}(\rho v)^{\mathrm{obj}}, \tag{3.86b}$$

$$f_7 = f_8 - \frac{1}{2}(f_3 - f_4) - \frac{1}{6}(\rho u)^{\mathrm{obj}} - \frac{1}{2}(\rho v)^{\mathrm{obj}}. \tag{3.86c}$$

The values ρ^{obj}, $(\rho u)^{\mathrm{obj}}$ and $(\rho v)^{\mathrm{obj}}$ change depending on the type of BC. They are known *a priori*. These values represent the restrictions imposed by each BC. The most usual cases are described in the following lines.

- **Moving wall:** Moving walls force the non-slip condition and impermeability. The velocity of the fluid is the same than the velocity of the wall $u^{\mathrm{obj}} = u_w$ and $v^{\mathrm{obj}} = v_w$. The density has to be corrected to accomplish mass conservation. First and second expressions of the equation system 3.85 relate the wall velocity normal to the surface with a variation of the objective density $\rho^{\mathrm{obj}} = [f_0 + f_3 + f_4 + 2(f_1 + f_5 + f_8)]/(1 + u_w)$.
- **Pressure inlet:** Pressure and density are related in LBM. Known pressure means known density $\rho^{\mathrm{obj}} = \rho^{\mathrm{in}}$. For this particular case and others when the equation for the first component of the momentum and the equation of the density are linear dependent the normal velocity is restricted. The example forces the first component to be $u^{\mathrm{in}} = [f_0 + f_3 + f_4 + 2(f_1 + f_5 + f_8)]/\rho^{\mathrm{in}} - 1$. In their article Zoe & He also make 0 the tangent velocity $v^{\mathrm{in}} = 0$.
- **Velocity inlet:** It is similar to pressure inlet. The redundancy between the density equation and the equation for the first momentum component links their values. The new density is given by the normal to the limit $\rho^{\mathrm{obj}} = [f_0 + f_3 + f_4 + 2(f_1 + f_5 + f_8)]/(1+u^{\mathrm{in}})$. The tangent component is independent to the rest of variables. Its effects can be included replacing directly $v^{\mathrm{obj}} = v^{\mathrm{in}}$.

3.3.4 Boundary Conditions as an optimisation process

A complete new set of BCs has been developed for the in-house new software LaBMoTer, an acronym of LAttice Boltzmann from MOtores TÉRmicos. The motivation to develop it is universality and robustness. Universality because every BC can be satisfied solving the same equations. Only some reference values and restrictions change. Robustness because the BCs can be

applied for every geometry. Some BCs can not accomplish multi-restrictive conditions in some special situations. For instance, when there is a vertical wall and only the components with the same horizontal speed participate, the equation for density and first momentum could be linear dependent. In this situation forcing density automatically fixes velocity. The equation system becomes over-determined. It happens when the number of restrictions is larger than the number of independent equations. The opposite and more usual situation is under-determined systems. The restrictions imposed by the BC can be satisfied by infinite combinations of the components of the incoming distribution. Both problems have been successfully resolved. Under-determined systems become determined when the Boundary Condition is solved as an optimisation problem. The inversion algorithm for over-determined systems returns the solution with lower quadratic error, exact values for determined systems.

The development process of new sets of BCs usually consists on imposing an arbitrary criteria to analyse their properties afterwords. In this thesis the process has been inverted. The criteria is fixed at the beginning and the BCs are the results. The procedure consists on an optimisation process where the function to be minimised represents the criterion mathematically. The BCs conditions are the constrains. The distribution components are the arguments to be optimised.

The selected optimisation method is the multipliers of Lagrange. The process starts with the definition of the minimisation function $\mathscr{F}(\boldsymbol{f})$. Typical minimisation functions are the square of the difference between the incoming distribution at the BC $\boldsymbol{f}^{\text{in}}$ with respect to other of reference $\boldsymbol{f}^{\text{ref}}$

$$\mathscr{F}(\boldsymbol{f}) = \sum_{\alpha} \frac{1}{2} \left(f_{\alpha}^{\text{in}} - f_{\alpha}^{\text{ref}} \right)^2 \tag{3.87}$$

or its equivalent corrected-weighted form

$$\mathscr{F}(\boldsymbol{f}) = \sum_{\alpha} \frac{1}{2w_{\alpha}} \left(f_{\alpha}^{\text{in}} - f_{\alpha}^{\text{ref}} \right)^2 . \tag{3.88}$$

Once the minimisation function has been established it is time to include the restrictions. Each restriction has to be a function equal to 0. Typical restrictions are known density (pressure) ρ^{in}

$$\mathscr{G}(\boldsymbol{f}) := \rho^{\text{in}} - \sum_{\alpha} f_{\alpha} = 0 \tag{3.89}$$

or known velocity $\vec{u}^{\,\text{in}}$

$$\mathscr{G}(\boldsymbol{f}) := \rho^{\text{in}} \vec{u}^{\,\text{in}} - \sum_{\alpha} \vec{c}_{\alpha} f_{\alpha} = 0. \tag{3.90}$$

Part of the distribution could be known *a priori*. The distribution is decomposed into the known part f_α^{kn} and the incoming part f_α^{in}

$$\sum_\alpha f_\alpha = \sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\in\Omega\}} f_\alpha^{\text{kn}} + \sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_\alpha^{\text{in}}. \tag{3.91}$$

The known portion is included in the equation system as part of the independent term. Once the restrictions have been established they are added to the objective function, each one multiplied by a new parameter λ called Lagrange multiplier

$$\mathscr{L}(\boldsymbol{f}^{\text{in}}, \boldsymbol{\lambda}) := \mathscr{F}(\boldsymbol{f}^{\text{in}}) + \sum_{k=1}^{N_R} \lambda_k \mathscr{G}_k(\boldsymbol{f}^{\text{in}}) \tag{3.92}$$

where k is the restriction index and N_R the number of restrictions. A local minimum exists when the variation respect each component of the distribution and each multiplier is 0,

$$\frac{\partial\mathscr{L}}{\partial f_\alpha^{\text{in}}} = 0 \quad \forall\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\} \qquad \text{and} \qquad \frac{\partial\mathscr{L}}{\partial\lambda_k} = 0 \quad \forall k. \tag{3.93}$$

The number of unknowns of the resultant equation system is the number of components of the distribution which participate in the BC Q^{in} plus the number of restrictions N_R. The most typical restrictions are in density and momentum (velocity). The weighted equation system written in the form $[\mathscr{M}]\boldsymbol{x} = \boldsymbol{b}$ for these restrictions looks like

$$[\mathscr{M}] = \begin{bmatrix} w_0^{-1} & 0 & 0 & \cdots & 1 & c_{x,0} & c_{y,0} & c_{z,0} \\ 0 & w_1^{-1} & 0 & \cdots & 1 & c_{x,1} & c_{y,1} & c_{z,1} \\ 0 & 0 & w_2^{-1} & \cdots & 1 & c_{x,2} & c_{y,2} & c_{z,2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 \\ c_{x,0} & c_{x,1} & c_{x,2} & \cdots & 0 & 0 & 0 & 0 \\ c_{y,0} & c_{y,1} & c_{y,2} & \cdots & 0 & 0 & 0 & 0 \\ c_{z,0} & c_{z,1} & c_{z,2} & \cdots & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{x} = \begin{Bmatrix} f_0^{\text{in}} \\ f_1^{\text{in}} \\ f_2^{\text{in}} \\ \vdots \\ \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{Bmatrix}, \qquad \boldsymbol{b} = \begin{Bmatrix} w_0^{-1} f_0^{\text{ref}} \\ w_1^{-1} f_1^{\text{ref}} \\ w_2^{-1} f_2^{\text{ref}} \\ \vdots \\ \rho^{\text{in}} - \sum f_\alpha^{\text{kn}} \\ (\rho u_x)^{\text{in}} - \sum f_\alpha^{\text{kn}} c_{x,\alpha} \\ (\rho u_y)^{\text{in}} - \sum f_\alpha^{\text{kn}} c_{y,\alpha} \\ (\rho u_z)^{\text{in}} - \sum f_\alpha^{\text{kn}} c_{z,\alpha} \end{Bmatrix}. \tag{3.94}$$

Every coefficient matrix of this type can be divided in 2×2 blocks. The block $[1,1]$ is a square sub-matrix $[\mathscr{F}]$ of size $Q^{\text{in}}\times Q^{\text{in}}$. It is the Hessian matrix of

the minimisation function $\mathscr{F}$

$$[\mathscr{F}]_{ij} = \frac{\partial^2 \mathscr{F}}{\partial f_i^{\text{in}} \partial f_j^{\text{in}}}. \tag{3.95}$$

The second block $[1,2]$ is a sub-matrix $[\mathscr{G}]$ of size $Q^{\text{in}} \times N_R$. It contains the derivatives of the restrictions with respect to each distribution component

$$[\mathscr{G}]_{ij} = \frac{\partial \mathscr{G}_j}{\partial f_i^{\text{in}}}. \tag{3.96}$$

The other blocks $[2,1]$ and $[2,2]$ are $[\mathscr{G}]^T$ and a matrix of zeros $[0]$ respectively. The resulting equation system is then

$$\begin{bmatrix} [\mathscr{F}] & [\mathscr{G}] \\ [\mathscr{G}]^T & [0] \end{bmatrix} \begin{Bmatrix} \boldsymbol{f}^{\text{in}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} [\mathscr{F}]\boldsymbol{f}^{\text{ref}} \\ \boldsymbol{m}^{\text{in}} \end{Bmatrix}. \tag{3.97}$$

where $\boldsymbol{m}^{\text{in}}$ are the first (density) and second moments (velocity) of the incoming distribution. The equation system can be simplified multiplying by

$$[\mathscr{M}] = \begin{bmatrix} [I] & [0] \\ -[\mathscr{G}]^T[\mathscr{F}]^{-1} & [I] \end{bmatrix} \tag{3.98}$$

resulting

$$\begin{bmatrix} [\mathscr{F}] & [\mathscr{G}] \\ [0] & -[\mathscr{G}]^T[\mathscr{F}]^{-1}[\mathscr{G}] \end{bmatrix} \begin{Bmatrix} \boldsymbol{f}^{\text{in}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} [\mathscr{F}]\boldsymbol{f}^{\text{ref}} \\ \boldsymbol{m}^{\text{in}} - \boldsymbol{m}^{\text{ref}} \end{Bmatrix}. \tag{3.99}$$

At this point the Lagrange multipliers are decoupled, and they can be solved directly. Two small equations systems replace the original

$$-[\mathscr{G}]^T[\mathscr{F}]^{-1}[\mathscr{G}]\boldsymbol{\lambda} = \boldsymbol{m}^{\text{in}} - \boldsymbol{m}^{\text{ref}} \qquad \text{and} \qquad [\mathscr{F}]\boldsymbol{f}^{\text{in}} = [\mathscr{F}]\boldsymbol{f}^{\text{ref}} - [\mathscr{G}]\boldsymbol{\lambda}. \tag{3.100}$$

Firstly the Lagrange multipliers are solved. Once their values are known they pass to the right hand side. Then the distribution can be calculated.

Implementation

The inner region is surrounded by the one-cell thick zone which corresponds with the BCs. Some of the components of the distribution at the boundary can not be streamed because the source or the destiny does not exist. The distribution which leaves the domain is lost whereas the distribution which comes into the domain has to be created. How the incoming population is depends on the type of BC and the population before streaming. The followed strategy consists on overlapping over the boundary nodes artificial distributions which stream into the domain. This overlapping distributions are called

incoming distributions f^{in}. The relation between the reference values (node of the domain) and the outer values (overlapping nodes) is given by expression 3.99. The selected minimisation function is the weighted one

$$\mathscr{F}(\boldsymbol{f}) = \sum_{\alpha} \frac{1}{2w_\alpha} \left(f_\alpha^{\text{in}} - f_\alpha^{\text{ref}}\right)^2, \tag{3.101}$$

whose Hessian matrix is $[\mathscr{F}]_{ij} = w_\alpha^{-1}\delta_{ij}$. The restrictions $[\mathscr{G}]$ are known density and first momentum. After building the coefficient matrix (and the premultiplicative matrix for the right hand side) the equation system 3.99 is solved. The resultant expression is the product of a matrix $[\mathscr{M}]$ and the vector which contains the reference distribution and the disparity of momenta

$$\{\boldsymbol{f}^{\text{in}}\} = [\mathscr{M}]\{\boldsymbol{f}^{\text{ref}},\, \rho^{\text{in}} - \rho^{\text{ref}}, (\rho\vec{u})^{\text{in}} - (\rho\vec{u})^{\text{ref}}\}. \tag{3.102}$$

The example proposed in figure 3.4 has the following equation system

$$\begin{bmatrix} 9 & 0 & 0 & 1 & -1 & 0 \\ 0 & 36 & 0 & 1 & -1 & 1 \\ 0 & 0 & 36 & 1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} f_2^{\text{in}} \\ f_6^{\text{in}} \\ f_7^{\text{in}} \\ \lambda_\rho \\ \lambda_{\rho u_x} \\ \lambda_{\rho u_y} \end{Bmatrix} = \begin{Bmatrix} 9f_2^{\text{ref}} \\ 36f_6^{\text{ref}} \\ 36f_7^{\text{ref}} \\ \rho^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} \\ \rho^{\text{in}} u_x^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} c_{x,\alpha} \\ \rho^{\text{in}} u_y^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} c_{y,\alpha} \end{Bmatrix} \tag{3.103}$$

The decoupled form of the same system is

$$\begin{bmatrix} 9 & 0 & 0 & 1 & -1 & 0 \\ 0 & 36 & 0 & 1 & -1 & 1 \\ 0 & 0 & 36 & 1 & -1 & -1 \\ 0 & 0 & 0 & -1/6 & 1/6 & 0 \\ 0 & 0 & 0 & 1/6 & -1/6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/18 \end{bmatrix} \begin{Bmatrix} f_2^{\text{in}} \\ f_6^{\text{in}} \\ f_7^{\text{in}} \\ \lambda_\rho \\ \lambda_{\rho u_x} \\ \lambda_{\rho u_y} \end{Bmatrix} \cdots$$

$$\cdots = \begin{Bmatrix} 9f_2^{\text{ref}} \\ 36f_6^{\text{ref}} \\ 36f_7^{\text{ref}} \\ \rho^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} - \sum\limits_{\alpha\in\{2,6,7\}} f_\alpha^{\text{ref}} \\ \rho^{\text{in}} u_x^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} c_{x,\alpha} - \sum\limits_{\alpha\in\{2,6,7\}} f_\alpha^{\text{ref}} c_{x,\alpha} \\ \rho^{\text{in}} u_y^{\text{in}} - \sum\limits_{\alpha\in\{0,1,3,4,5,8\}} f_\alpha^{\text{kn}} c_{y,\alpha} - \sum\limits_{\alpha\in\{2,6,7\}} f_\alpha^{\text{ref}} c_{y,\alpha} \end{Bmatrix}. \tag{3.104}$$

It can be seen how λ_ρ and $\lambda_{\rho u_x}$ are linear dependent. The equation system becomes under-determined and inconsistent. There are several strategies to apply when it happens. The algorithm which solves the equation is robust. Over-restricted equations are resolved returning the solution with minimum euclidean norm of the residual. When the density and the horizontal velocity are consistent the error is zero. The first strategy is then forcing one of the previous variables to be consistent with the restriction imposed by the BC. This is the same strategy followed by Zoe & He to define their BCs. The second option is to include the central (or more) component of the distribution into the system. The reference value for this component should be $f_0^{\text{ref}} = f_0^{\text{kn}}$. The inconsistency between λ_ρ and $\lambda_{\rho u_x}$ is absorbed by this component. The BC is achieved but with modification of non-incoming populations. Both previous strategies can be applied together. If he original equation system is consistent the central distribution does not change.

ρ^{in} and $\vec{u}^{\text{in}}$ fix the BC type. They are the restrictions and they are known at the beginning. The reference distribution f_α^{ref} is used for tuning. Some spacial cases returns the already known BC developed by other researchers.

Moving wall: modified bounce-back

In this thesis the bounce-back formulation has been extended to simulate moving walls. A particularity of moving walls in Lattice Boltzmann Methods is that they can inject mass into the domain for normal velocities. In classical CFD solvers like Finite Volumes the mesh is readjusted to be coherent with the displaced geometry. On the contrary the lattice is fixed. It does not move neither readjust the control volume. Only the nodes, discrete locations, can be enabled and disabled in agreement with its immersion in the fluid phase.

The new BC has been developed following the optimisation procedure. The restrictions differs from the generic form proposed in equation 3.3.4. Instead of working with density and velocity, the wall forces mass and momentum flux. As usual, the distribution can be divided in known f_α^{kn} and unknown f_α^{in} components. The unknown components of the distribution f_α^{in} come into the domain, so all of them contribute to the incoming fluxes. The outgoing fluxes are associated to some of the known components f_α^{kn}. Every incoming component f_α^{in} has its opposite outgoing counterpart $f_{\bar{\alpha}}^{\text{kn}}$. The bar over the index means opposite direction. The associated velocities satisfy the equality $\vec{c}_\alpha = -\vec{c}_{\bar{\alpha}}$.

The restrictions to be considered are two. One of them is impermeability. The mass flow rate through the wall has to be zero. The mass flow rate which

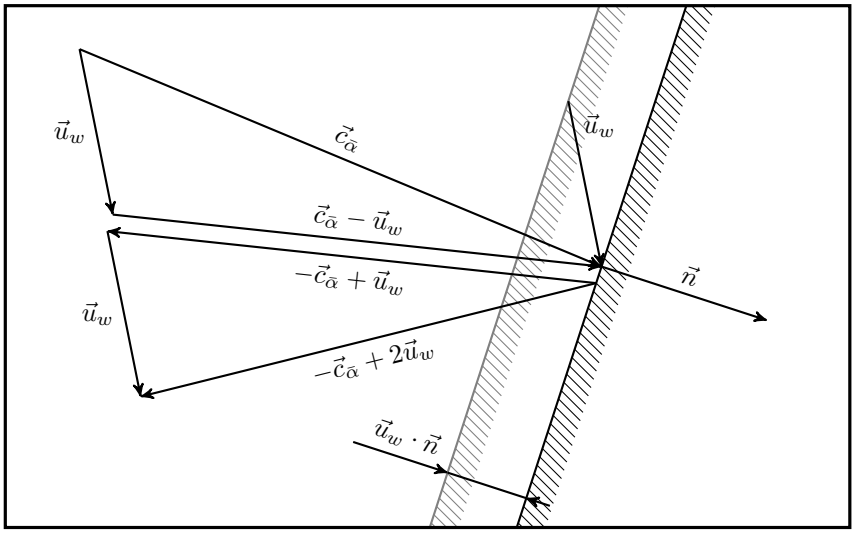

Figure 3.5: Geometrical representation of the bounce-back boundary condition. The particles with velocity $\vec{c}_{\bar{\alpha}}$ has a relative velocity with respect to the wall $\vec{c}_{\bar{\alpha}} - \vec{u}_w$. They return, in relative speed, with same velocity but with opposite direction. In absolute magnitudes their velocity after collide is $-\vec{c}_{\bar{\alpha}} + 2\vec{u}_w$. The wall sweeps an area equal to $\vec{u_w} \cdot \vec{n}$.

leaves the domain through the wall per unit area and unit time is

$$\dot{m}^{\text{in}} = \sum_{\alpha \in \{\alpha | \vec{x} - \Delta\vec{x}_{\bar{\alpha}} \notin \Omega\}} f_{\bar{\alpha}}^{\text{kn}} \left[(\vec{c}_{\bar{\alpha}} - \vec{u}_w) \cdot \vec{n} \right]. \tag{3.105}$$

The incoming flow has to be equal to satisfy the impermeability restriction, so

$$\mathscr{G}_0 := \sum_{\alpha \in \{\alpha | \vec{x} - \Delta\vec{x}_{\bar{\alpha}} \notin \Omega\}} f_{\alpha}^{\text{in}} \vec{c}_{\alpha} \cdot \vec{n} - \sum_{\alpha \in \{\alpha | \vec{x} - \Delta\vec{x}_{\bar{\alpha}} \notin \Omega\}} f_{\bar{\alpha}}^{\text{kn}} \left[(\vec{c}_{\bar{\alpha}} - \vec{u}_w) \cdot \vec{n} \right] = 0. \tag{3.106}$$

This is the first restriction to be satisfied.

The second restriction is known velocity. When a particle with velocity $\vec{c}_{\bar{\alpha}}$ impinges the wall it is bounced-back at $\vec{c}_{\alpha} + 2\vec{u}_w$ (see figure 3.5). Taking the average in time the mean speed is $(\vec{c}_{\bar{\alpha}} + \vec{c}_{\alpha} + 2\vec{u}_w)/2 = \vec{u}_w$. In other words, the mean velocity satisfies the non-slip condition. The incoming momentum flux is then

$$\vec{p}^{\text{in}} = \sum_{\alpha \in \{\alpha | \vec{x} - \Delta\vec{x}_{\bar{\alpha}} \notin \Omega\}} f_{\bar{\alpha}}^{\text{kn}} \left(\vec{c}_{\bar{\alpha}} - 2\vec{u}_w \right) \left[(\vec{c}_{\bar{\alpha}} - \vec{u}_w) \cdot \vec{n} \right]. \tag{3.107}$$

The increment in speed is replaced in the lattice by an equivalent increment

in mass to get the same momentum

$$\sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_\alpha^{\text{in}}\vec{c}_\alpha(\vec{c}_\alpha\cdot\vec{n}) - \sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_{\bar{\alpha}}^{\text{kn}}\left(\vec{c}_{\bar{\alpha}}-2\vec{u}_w\right)\left[(\vec{c}_{\bar{\alpha}}-\vec{u}_w)\cdot\vec{n}\right]=0. \tag{3.108}$$

The next step is to select the reference values. To be coherent with pure bounce-back non-moving walls the reference values are the opposite distribution $f_\alpha^{\text{ref}} = f_{\bar{\alpha}}^{\text{kn}}$. Its physical meaning is the minimisation of the difference between the inverse of the known distribution and the incoming distribution.

Free slip boundary condition

Free slip walls are free of drag forces. It is not usual in nature to find limits like this. However they have a lot of practical applications. It is recommended to delimit the domain with free slip walls in these situations:

- **Symmetry.** Every net flux through a symmetric limit is zero. The same happens with slip walls. Symmetry is completely equivalent to a straight free slip wall.
- **Flow confinement.** Some times it is needed to change the section or the incident angle of the fluid flow without loss of mass. A good way to do it without loss of energy is to channel the domain with free slip walls. The flow can be reoriented by curved channels or a cascade of curved blades. In similar way the flux can be accelerated by the reduction of the cross section.
- **Free of drag walls.** Slip walls are not common in nature. However could be interesting to assume this hypothesis in same special situations.

The procedure to calculate free slip walls is practically equal to the procedure to calculate non-slip walls. The restriction about permeability is the same. The mass flow rate through the free slip wall is zero

$$\sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_\alpha^{\text{in}}\vec{c}_\alpha\cdot\vec{n} - \sum f_{\bar{\alpha}}^{\text{kn}}\left[(\vec{c}_{\bar{\alpha}}-\vec{u}_w)\cdot\vec{n}\right]=0. \tag{3.109}$$

The unique difference is the assumption about velocity. The particles with velocity $\vec{c}_{\bar{\alpha}}$ has a relative velocity with respect to the wall $\vec{c}_{\bar{\alpha}}-\vec{u}_w$. They are specularly reflected after strike the wall. In absolute magnitudes their velocity after collide is $\vec{c}_{\bar{\alpha}}-2[(\vec{c}_{\bar{\alpha}}-\vec{u}_w)\cdot\vec{n}]\vec{n}$. The wall sweeps an area equal to $\vec{u_w}\cdot\vec{n}$. The free slip wall forces the normal velocity to be the same and keeps the tangential velocity (see figure 3.6)

$$\sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_\alpha^{\text{in}}\vec{c}_\alpha(\vec{c}_\alpha\cdot\vec{n}) - \sum_{\alpha\in\{\alpha|\vec{x}-\Delta\vec{x}_{\bar{\alpha}}\notin\Omega\}} f_{\bar{\alpha}}^{\text{kn}}\left[\vec{c}_{\bar{\alpha}}-2\left[(\vec{c}_{\bar{\alpha}}-\vec{u}_w)\cdot\vec{n}\right]\vec{n}\right]\left[(\vec{c}_{\bar{\alpha}}-\vec{u}_w)\cdot\vec{n}\right]=0. \tag{3.110}$$

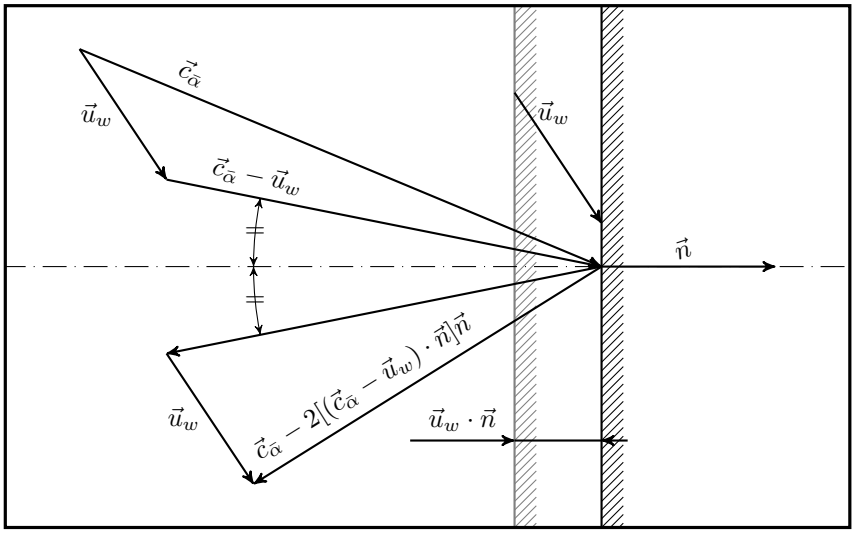

Figure 3.6: Geometrical representation of the free slip boundary condition.

The reference values are the distribution before streaming $f_\alpha^{\text{ref}} = f_\alpha^{\text{kn}}$. Other references can be selected.

Pressure inlet/outlet

Known pressure (or density in LBM) is one of the most usual BCs. Their used is recommended for:

- **Driving force.** Two different pressure BCs in different locations create pressure gradient. It drives the fluid to flow from the high pressure BC to the low pressure BC. Limiting the domain with pressure BCs is highly recommended when it is expected high pressure gradients in the solution. An example is large constrains, such as porous media. All the flow have to pass through a little cross section. When the flux is forced by the BC, the speed at these points could be extremely large. As consequence the simulation could diverge.

- **In combination with velocity inlet/outlet.** When there is a BC which fixes net incoming flux (velocity inlet) is highly recommended the existence of at least one pressure inlet or outlet. A domain completely surrounded by fixed fluxes is over-restricted. There is a cumulative term. Consequently the simulation diverges.

- **Outlet in external aerodynamics.** In external aerodynamics the velocity of the immersed body is known. The incoming flux is usually imposed by a velocity inlet BC upstream, where the perturbations of the immersed body are weak. To avoid over-restrictions the outlet, located downstream, is a pressure outlet BC.

In LBM pressure is proportional to the lattice density. It means that at the pressure inlet the density is known ρ^{in}. The velocity is a free parameter. Several strategies can be applied. Copy the velocity of the cell before streaming $\vec{u}^{\text{in}} = \sum_\alpha f_\alpha \vec{c}_\alpha / \sum_\alpha f_\alpha$ approximates zero gradient velocity. The reference values can be the existent distribution $f_\alpha^{\text{ref}} = f_\alpha$ or the equilibrium $f_\alpha^{\text{ref}} = f_\alpha^{\text{eq}}$. The generic expression 3.3.4 solves the distribution.

Velocity inlet/outlet

Known velocity is another of the usual BCs. Their application is recommended for

- **Far field.** In external aerodynamics forcing velocity upstream is the best solution. It has to be located far enough to avoid being perturbed by the presence of the immersed body.
- **Forcing mass flow rate through a section.** Known mass flow rate through a cross section can be established forcing the mean velocity for every cell in this area. This BC is very useful for internal aerodynamics. Sometimes the domain is enlarged to get full-developed velocity profiles. LBM is quasi-incompressible. It means that the density can change slightly. Consequently forcing the velocity does not directly fix the mass flow rate. There is a small error of the same order than the velocity in lattice units.

Net mass flow rate can be driven by two couples of BC. Pressure drop (two BCs with known pressure) and velocity inlet with pressure outlet. Pressure drop is the appropriate solution for internal flows with large constrictions. Otherwise the constrain could induce extremely high values of pressure (density) upstream and/or extremely low pressure downstream the obstacle. On the contrary, when there are not any important restriction, moving the flow by means of the pressure drop could produce extremely high velocities. When it is not expected any important pressure drop the best option is to use a mix of velocity inlet and pressure outlet.

The generic equation 3.3.4 to solve BCs is used to impose velocity at the boundary. The incoming velocity $\vec{u}^{\text{in}}$ is known. The pressure can be estab-

lished by different ways. The most usual is zero gradient, when $\rho^{\text{in}} = \sum_\alpha f_\alpha$. The reference values can be the existent distribution $f_\alpha^{\text{ref}} = f_\alpha$ or the equilibrium $f_\alpha^{\text{ref}} = f_\alpha^{\text{eq}}$.

Total pressure boundary condition

Total pressure BC is a special case of pressure inlet where the velocity field and the pressure field are coupled. The total pressure P_0^{Ph} for incompressible flow is given by the Bernoulli's equation

$$P_0^{\text{Ph}} = P^{\text{Ph}} + \frac{1}{2}\rho^{\text{Ph}}\|\vec{u}^{\,\text{Ph}}\|^2. \tag{3.111}$$

This equation can be converted to lattice units resulting

$$\rho_0 = \rho\left(1 + \frac{\|\vec{u}\|^2}{2c_s^2}\right). \tag{3.112}$$

For the total pressure BC the value of ρ_0 is known. From here different strategies can be applied. The most usual is to keep the velocity of the cell $\vec{u}^{\,\text{in}} = \sum_\alpha f_\alpha \vec{c}_\alpha / \sum_\alpha f_\alpha$. The new density is computed $\rho^{\text{in}} = \rho_0/(1+\|\vec{u}\|^2/2c_s^2)$. The distribution is obtained after applying equation 3.3.4. The reference values can be the existent distribution $f_\alpha^{\text{ref}} = f_\alpha$ or the equilibrium $f_\alpha^{\text{ref}} = f_\alpha^{\text{eq}}$.

This BC can be used in replacement of the pressure inlet BC. It is more robust against perturbations.

Pressure and velocity boundary condition

Both pressure, density in lattice units ρ^{in}, and velocity $\vec{u}^{\,\text{in}}$ can be fixed at the inlet. The generic equation 3.3.4 can be applied to get the resultant distribution without any previous calculation. The reference values can be the existent distribution $f_\alpha^{\text{ref}} = f_\alpha$ or the equilibrium $f_\alpha^{\text{ref}} = f_\alpha^{\text{eq}}$. This BC is very restrictive. Consequently, to avoid over-restrictions, two degrees of freedom has to be included in the other limits. They can be introduced by a mix of two different BCs (pressure and velocity BCs) or by a single zero gradient BC.

Zero gradient

Zero gradient BC is approximated by a direct copy of the population before streaming $f_\alpha^{\text{in}} = f_\alpha$. Interpolation can be used instead to get ρ^{in} and $\vec{u}^{\,\text{in}}$ and to improve the accuracy. Its use is not recommended. This BC is weakly stable. Its application can be considered for full-developed fluxes in downstream direction.

3.3.5 High order boundary conditions

All the previous BCs admit higher order approximations. A more expensive and complicated formulation is able to capture the curvature of the walls and approximate better their location. There is only one case when applying high order could have a boomerang effect, the corners. A corner have infinite curvature. As consequence improvement in accuracy is not guaranteed. Acicular mullite, the material to be analysed in this thesis, is one of these exceptions. The crystals are sharp needles with multiple edges.

The BC proposed by Guo, Zheng and Shi [11] is second order accurate. They decomposed the distribution into the equilibrium and non-equilibrium parts. A fictitious node enforces the condition at the wall whereas the non-equilibrium components are extrapolated by a first order tendency. Bouzidi, Firdaouss and Lallemand [13] implement a similar technique for BGK operator and moving walls. The formulation is a modification of bounce-back BC. Instead of forcing the location of the bounced distribution to be coincident with lattice nodes the position after streaming is generic. In a second stage of the streaming step the population is redistributed among the surroundings nodes.

3.4 Grid refinement

LBM works very well with uniform lattice. However uniform lattice is not practical when the geometry or other phenomena present multiple scales. To capture the detail it is required fine lattice. However large volumes of fluid without any perturbation can be successfully solved with coarser meshes. Consequently discretising whole the domain with the same lattice spacing is not optimum. The lattice can be refined to be adapted to the required level of detail. Several refinement techniques have been developed. Usually one or more overlapping regions coexist in the interface between lattices. This is the case of the algorithm developed by Neumann and Neckel [14]. It combines refinement with data arrangement to optimise memory locality. The framework Peano [15] provides the second capability. Usually accurate interfaces between lattices of different spacing requires interpolation. This task is computationally expensive. Geier, Greiner and Korvink [16] propose to use bubble functions for interpolation to obtain maximum performance. Their algorithm is second order accurate. Other authors like Rohde *et al.* [17] use extensive magnitudes. Their work is focused on reducing computational effort. They argue that extensive magnitudes avoid interpolation and scaling operations. The conservation is obtained in a natural way. The negative counterpart is penalty in accuracy. Yu, Mai and Shyy [18] propose to discretise in blocks instead of refining locally.

Arranged data are cache friendly and speed up the calculation in comparison with random access. The refinement factor is generic. Each block can be in contact with multiple lattices with different lattice spacing. Filippova and Hänel [19] propose local refinement for BGK model. The transference between lattices is realised across overlapping nodes. The single adjustable parameters of BGK model is used to keep viscosity constant between lattices. They non-equilibrium distribution has to be corrected during the transference from one lattice to other. Similarly Chen *et al.* [20] transfer the distribution between lattices. The difference is the formulation. They use extensive magnitudes instead of intensive like density to avoid the conversion operations. Mass conservation is directly achieved using this formulation. Dupuis and Chopard [21] also focus their research in performance improvement. The strategy is based on simplifying the calculation in comparison with other techniques.

Several requirements are necessary for compatibility between lattices. All the physical properties have to be equal. The elapsed physical time from the beginning has to be the same in both lattices. The particles have to travel at the same physical speed to cover the same physical distance during the same physical time step. The populations have to travel from one lattice to other. The transference could be established by interpolation in time and space. This strategy makes the simulation very expensive. The single way to avoid interpolation is by integer refinement factors $K \in \mathbb{N}$. The immediately coarser lattice spacing is K times larger than the fine lattice $\Delta x_c^{\mathrm{Ph}} = K\Delta x_c^{\mathrm{Ph}}$. The velocity in lattice units is forced to be the same. As a consequence the time step also scales with the lattice $\Delta t_c^{\mathrm{Ph}} = K\Delta t_f^{\mathrm{Ph}}$. In other words, K time steps of the finest mesh elapses the same physical time than a single one of the coarsest lattice. This proportion between time steps has to be conserved during the simulation.

Now it is time to see the relation between the lattice magnitudes in both scales. It is already known how $t_c = Kt_f$ and $\vec{x}_c = K\vec{x}_f$, whereas the velocity and the density are the same $\vec{u}_c = \vec{u}_f$ and $\rho_c = \rho_f$. The NS equation for the coarsest scale can be scaled to the magnitude of the finest scale

$$\begin{aligned}\partial_{t_f}(\rho\vec{u}) + \vec{\nabla}_{\vec{x}_f}\cdot(\rho\vec{u}\otimes\vec{u}) = -\vec{\nabla}_{\vec{x}_f}\cdot(\rho\theta[I])\cdots \\ \cdots + \vec{\nabla}_{\vec{x}_f}\cdot\left[\frac{\theta}{K}\left(\tau_c - \frac{\Delta t}{2}\right)\left(\vec{\nabla}_{\vec{x}_f}\otimes\rho\vec{u} + (\vec{\nabla}_{\vec{x}_f}\otimes\rho\vec{u})^T\right)\right]. \quad (3.113)\end{aligned}$$

Only the factor related with momentum diffusion changes. This factor can be written in terms of the thinnest lattice. Both have to be equal

$$\frac{\theta}{K}\left(\tau_c - \frac{\Delta t}{2}\right) = \theta\left(\tau_f - \frac{\Delta t}{2}\right). \quad (3.114)$$

The relaxation time of the fine mesh can be written in terms of the relaxation

time for the coarse lattice and the refinement factor

$$\tau_f = \frac{1}{K}\left(\tau_c - \frac{\Delta t}{2}\right) + \frac{\Delta t}{2}. \tag{3.115}$$

Equal velocity and density for both scales automatically fixes equal equilibrium distribution. However the non-equilibrium parts are different. Only the second momentum is different to 0. It appears as $\Pi^{(1)} = -\theta\tau\left(\vec{\nabla}\otimes\rho\vec{u} + (\vec{\nabla}\otimes\rho\vec{u})^T\right)$. The quotient for each component is

$$\frac{\Pi_c^{(1)}}{\Pi_f^{(1)}} = \frac{\tau_c\left(\vec{\nabla}_{\vec{x}_c}\otimes\rho\vec{u} + (\vec{\nabla}_{\vec{x}_c}\otimes\rho\vec{u})^T\right)}{\tau_f\left(\vec{\nabla}_{\vec{x}_f}\otimes\rho\vec{u} + (\vec{\nabla}_{\vec{x}_f}\otimes\rho\vec{u})^T\right)}. \tag{3.116}$$

The relation between gradient operators scales with the physical lattice spacing $K\vec{\nabla}_{\vec{x}_c} = \vec{\nabla}_{\vec{x}_f}$. With ρ and $\vec{u}$ equal for both scales the previous equation is simplified to

$$\frac{\Pi_c^{(1)}}{\Pi_f^{(1)}} = \frac{\tau_c}{\tau_f K}. \tag{3.117}$$

This relation is satisfied if each non-equilibrium component also scales with the same relation

$$\frac{f_{c,\alpha}^{\text{neq}}}{f_{f,\alpha}^{\text{neq}}} = \frac{\tau_c}{\tau_f K}. \tag{3.118}$$

This method is used by Dupuis and Chopard [21]. Filippova and Hänel [19] proposes an equivalent relation between relaxation times and non-equilibrium parts.

After define this relation the equivalent collision steps have been successfully found. It is time to fix the stream step. There are two points of view for streaming from one lattice to other (see figure 3.7). LBM can be seen as a volumetric formulation. Every node is surrounded by a volume. When the volume is divided the centroids of each sub-volume are not necessarily coincident with the parent centroid. Each volume is divided in an integer number of sub-volumes. The population of the parent cell is distribute among the descendent ones.

The second alternative is the nodal formulation. The nodes of the coarse lattice are coincident with some of the nodes of the fine lattice. Extra nodes, whose values are interpolated, are added along the frontier.

Each method has its own advantages and disadvantages. In volumetric formulation, after refine the lattice each node of the coarse mesh is associated to K^D nodes of the fine mesh, where D is the dimensionality. The mass and momentum contained in the parent cell is directly transferred to the descendants and *vice versa*. Without any advance interpolation locality is maintained. On

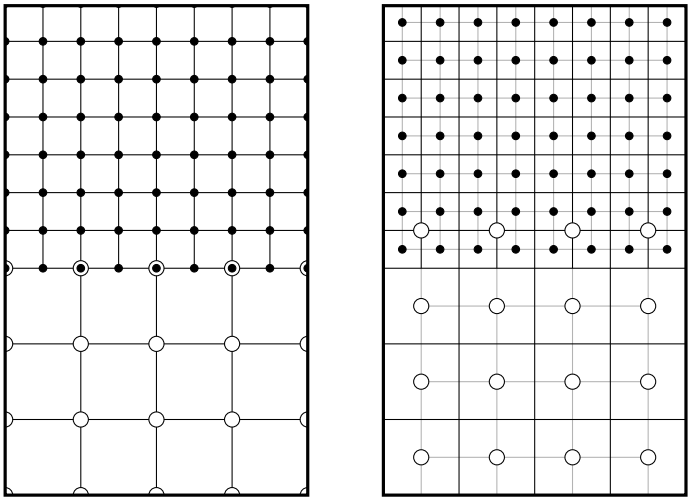

Figure 3.7: (left) Nodal mode. (right) Volumetric mode.

the contrary, nodal formulation is not totally local. Interpolation recovers data from multiple surrounding nodes. As a consequence, advance algorithms are required to ensure mass conservation. The number of nodes after refinement also reflects the difference. It can be observed in figure 3.7 how each divided cell in volumetric formulation is always associated to the same number of children cells. On the contrary, the number of new nodes in nodal mode is not intuitive. The simplest example, a flat interface, only adds new nodes in between. The coarse nodes at the limits are only surrounded by a single fine node in the direction of the interface. Consequently after refinement the line where both lattices overlap contains $K(N-1)+1$ nodes of the fine cells. The difference is larger if the frontier does not follow a straight line.

On the contrary interpolation is easier for nodal mode. The values at the hanging nodes are weighted sums of the surrounding coarse nodes. In volumetric mode the populations are directly transferred. If no interpolation is considered then every children node contains the same distribution. Better accuracy can be achieved if the fine nodes follows the tendency of the coarse lattice.

LaBMoTer, the software developed in Centro de Mantenimiento de Transporte (CMT-Motores Térmicos) in the frame of this PhD, redistribute the components following their first gradient. It is called the Piecewise Linear Method (PLM). Its task is to reconstruct the continuous flow field from finite volumes. This reconstruction technique is applied for each component of the

distribution separately

$$f_\alpha^f = f_\alpha^c + (\vec{x}_\alpha^f - \vec{x}_\alpha^c) \cdot \vec{\nabla} f_\alpha^c. \tag{3.119}$$

Higher polynomials and other techniques can be used instead of improving the interpolation. One problem of this method is the possibility of overestimating the value of the gradient. Couples of fine cells touching the coarse grid alternates over-estimation and under-estimation of their values.

3.5 Software

Several solvers based on LBM have been developed or used for different purposes. In this section these solvers are briefly described. It is explained the reasons why they have been developed as well as their characteristics.

3.5.1 Test environment

The test environment is a D2Q9 solver exclusively created for development purposes. 2D simulations are run with this application for validation, measurement of the performance and robustness testing. Every new advance in BCs and models has been tested in this environment. The implementation can be also optimised by a collection of benchmarks.

A test environment has to be fast and simple. Simplicity is obtained by a modular design. Every function is created to be as independent as possible. This strategy facilitates the incorporation, modification or replacement of subroutines. By default the test environment collides and streams whole the domain. There is periodicity for every direction. The streamed and the collided populations for the same time step are in memory. The BCs can be calculated from both distributions. They are applied afterwards, over-writting the populations before being collided again.

Modular design is easy using C++. Every modulus is equivalent to a class. In addition C++ also satisfies the second criterion, fast execution. This is the reason why the test environment is written in this language. But there is a candidate still better to run small simulations. The workstation is equipped with a C2070 GPU. This device is manufactured by nVidia Corporation. They also provides for free a library to run applications on the GPU, what is called GPGPU. The name of this library is CUDA, an acronym of Compute Unified Device Architecture. Running the simulations on the GPU accelerates the calculation in more than an order of magnitude. It is a technique widely extended in HPC.

Thanks to this design every new improvement can be easily incorporated and tested in few minutes. Just to have a general idea. The waLBerla framework, designed from the beginning for HPC applications, can solve five million cells per second and core. The workstation can refresh around 120M cells per second in total (D3Q19 scheme). The test environment, without being exhaustively optimised, refreshes up to 495M cells per second (D2Q9 scheme). Only memory alignment has been considered for optimisation. Random access has an important penalty in performance in GPGPU applications.

The most usual case for testing LBM is the driven cavity since the simulation of Hou *et al.* [22].

3.5.2 LaBMoTer

LaBMoTer is an acronym of Lattice Boltzmann from Motores Térmicos. It is a CFD solver based on LBM. It works with D3Q19 scheme. The objective of this software was to be able to run a simulation of a sample of porous media (filter) in the workstation. The simulation of porous media demands large amounts of memory. In one hand, the smallest details such as interstices have to be captured. In the other hand, the domain has to be large enough to contain several of the greatest structures. This two scales are very different. There is a big domain finely discretised. This is the reason for memory demand.

Before starting with the development two different opposed concepts have to be taken into consideration, the objective and the limitations. The objective is to simulate a filter. Its geometry is very complex and it demands a lot of memory. The limitations are mainly determined by the computer. It is a workstation with 128Gb of RAM and 24 threads (12 cores with multi-threading technology). To take advantage of all the processor the code has to run in parallel. For shared memory architectures, such as the workstation, a fast and easy option to parallelise is OpenMP library.

With respect to the memory the following actions were considered to reduce the demand as much as possible. The most important one is adaptive lattice. It can save a lot of memory. At the beginning the coarse grid occupies whole the domain. Accordingly to a criterion the cells can be divided into others in a recursive way (volumetric formulation, see section 3.4). This strategy can be applied, for instance, to capture the details of the geometry. Fine lattice is used only where it is needed. One important characteristic is that LaBMoTer can adapt the lattice on the fly. Typical criteria are high gradient (in cell) of a variable, threshold value or geometry. All of them have been implemented. Mesh adaptation is the best way to reduce memory consumption without loss accuracy.

However this methodology has a negative counterpart. When a cell is divided eight more are created. They are located at the end of an array. Consequently memory alignment is broken. The location of the children cells become chaotic. Indexes of neighbour nodes can not be calculated by arithmetic any more. Only the first level is arranged. The children cells are indexed to their parent an *vice versa*. The construction of the neighbourhood starts from the first level. The next level is connected by the indexes to the children. Successive neighbours for different levels are found following the index to the correct descendant. This is done in a recursive way. Each descendant can be divided and have more descendants. Once the appropriate neighbourhood has been calculated two different strategies can be applied. Each cell can be linked to the surroundings by storing the indexes. The neighbourhood has to be calculated only once. After that the stored links can be used to access directly to the surrounding nodes. Storing the indexes demands as much memory as a distribution. The second option is to calculate the indexes every time. It saves the memory required to storage them. However a penalty in performance has to be paid. Firstly the number of arithmetic operations increases. The computational time increases accordingly. Secondly the indexes to children are not aligned. It produces a lot of cache misses. Moving data is much more expensive (not only in time, but also in energy consumption) than arithmetic. Because the limiting factor is memory LaBMoTer calculates the indexes each iteration. It drastically slows the calculation down. waLBerla, which uses arranged storage, is up to 10 times faster than LaBMoTer.

3.5.3 waLBerla

waLBerla is the acronym of widely applicable Lattice Boltzmann from Erlangen. It is a general purpose framework originally developed to work with LBM. It can be used for other purposes like heat transfer. Its highest priority is performance and scalability. At the same time the framework has to be easy to use, modular and extensible. Each intensive differentiable task is executed by a different piece of code called kernel. Different versions for the same task exist. Each version contains a specific algorithm to adapt the solution to the requirements. Accuracy, robustness and convergence are the typical criteria. Computational time is the most usual magnitude to be minimised (maximum performance). Different kernels solves the same physics with different degree of accuracy, robustness and stability. In addition the algorithm has to be adapted to the computer architecture to get maximum performance. The performance of the algorithm can be sensible to the size of the problem too. Diving the solver in kernels facilitates the adaptation to multiple scenarios (machine architecture, size, accuracy...).

On the contrary to LaBMoTer, waLBerla has been designed to be run

on the most powerful supercomputers [6]. Consequently high scalability and performance are mandatory. The best performance is obtained for arranged data. The data demanded for computations can be prefetched when their location in memory is predictable. Prefeching consists on overlapping the calculations with data transferences. This technique minimises cache misses and improves performance. waLBerla divides the domain in blocks. Each block is a rectangular cuboid. Every cuboid contains a single lattice size. This strategy provides arrange data for every block. However the lattice size can not be adapted locally. What is done in practice is to refine the block entirely. It is a compromise between refining the entire domain and local adaptation. Full fine mesh consumes a lot of memory. Local adaptation deteriorates the performance.

waLBerla is prepared to work in the largest supercomputers. These machines have distributed memory. The best option for parallelise the code are Message Passing Interface (MPI) libraries. This is a second reason to divide the domain in blocks. The data of each block is only accessible by a single core. High parallel programs have to distribute the workload among the computational resources in agreement with their characteristics to obtain maximum performance. The computational load of each resource should be balanced. In homogeneous systems the workload should be the same for every resource. When a block is refined the number of nodes in it increases. It breaks the good load balancing. The solution is to divide the block in a recursive way. When a block is refined it is also divided. Each children block is a scaled version of the parent. Every block always contains the same number of nodes and all of them are equal in shape. Homogeneous load balancing is reached.

The good capabilities of waLBerla have been widely demonstrated. Several publications corroborates that such as [6]. Visit the waLBerla web page for more information.

3.6 Validation

3.6.1 3D driven-cavity

LaBMoTer has been validated with two different cases. Both of them are usually employed for this purposed or for calibration. One of them is the 3D driven-cavity. The solution of this problem is well known. A lot of bibliography of the solution exists, including the solution of DNS. Iwatsu *et al.* [23] dive a complete description of the structures of cubic 3D driven cavity for several Reynolds numbers. In a similar way Albensoede and Kuhlmann [24] calculate several 3D driven cavities for multiple aspect ratio. Their solver based on

spectral methods (Chebyshev-collocation) is validated up to Reynolds 1000. Leriche and Gavrilakis [25] also use Chebyshev-collocation methods. They calculate cubic driven cavities for several Reynolds.

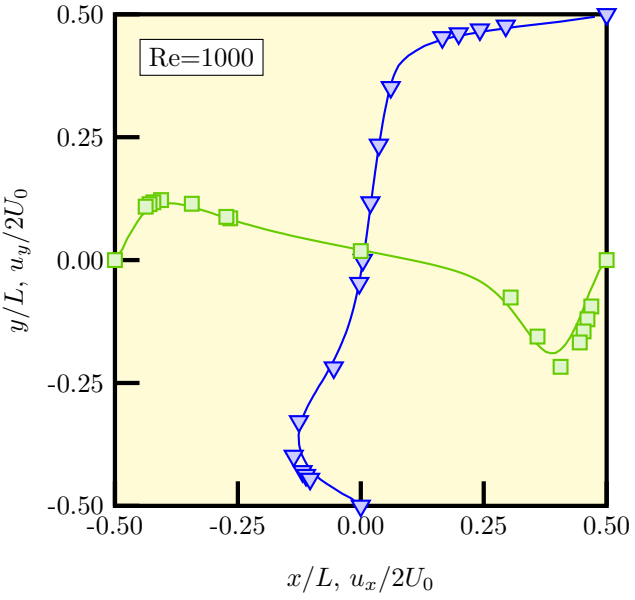

Figure 3.8: Dimensionless velocity components u_y and u_x for axis $(x, 0, 0)$ and $(0, y, 0)$ respectively in 3D driven cavity at Re=1000.

Driven cavity is a purely academic problem. It has no application on industry. The simulation calculates confined fluid in a cubic box where one of the walls moves at constant velocity and the rest are stopped. The Reynolds number it is defined from the velocity of the moving wall and the size of the domain. In this particular case it has been selected Re=1000. Considering the domain as a unitary box of size 1x1x1 and the origin of coordinates located in the middle of the domain, several proves have been located along the axis $(x, 0, 0)$ and $(0, y, 0)$. The second component of the velocity has been measured for the first axis and the first component of the velocity for the second. Both the results and the reference values are practically coincident 3.8.

3.6.2 Circular cylinder in cross-flow

The circular cylinder with cross flow is the second case. As well as for the driven cavity, a lot of bibliography exist about it. Another advantage are the measurements. Circular cylinder in cross flow is one of the most studied

cases in fluid dynamics. The flow field around it is something very easy to be measured. For moderate Reynolds an instability appears passing the cylinder. Alternate vortexes are detached creating the von Kármán vortex street. The frequency of detaching of the vortexes is known and it is correlated with the Reynolds number. A dimensionless number relates the frequency, the characteristic length and the characteristic speed, the Strouhal number

$$St = \frac{fD}{U_\infty}. \tag{3.120}$$

The model is considered valid if the Strouhal number obtained in the simulation approximates the measured values.

The simulation has been realised for Reynolds number equal to 250. The period (and then the frequency) can be calculated comparing the solution at two different times. One period is proportional to the distance between similar structures. To convert distance to time it is needed to know the velocity of the structures. This velocity can be easily obtained measuring the displacement of a structure during a known time 3.9 The resultant Strouhal number after

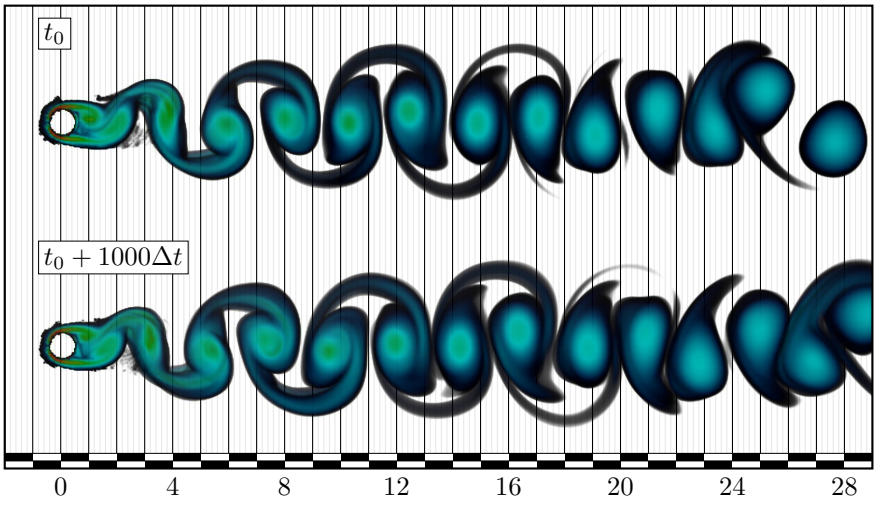

Figure 3.9: Flow field for an immersed circular cylinder. Volumetric render of the vorticity in logarithmic scale after 15000 and 16000 iterations. Units in diameters.

calculating the period and the frequency is 0.205. This value approximates very well the expected value. The solver passes the validation test successfully.

Bibliography

[1] He, X.Y.; Luo, L.S. A priori derivation of the lattice boltzmann equation. *PHYSICAL REVIEW E*, 55(6, A):(1997), –6333.

[2] Freitas, R.K.; Henze, A.; Meinke, M.; Schroeder, W. Analysis of lattice-boltzmann methods for internal flows. *COMPUTERS & FLUIDS*, 47(1):(2011), 115–121.

[3] Yu, H.; Girimaji, S.S.; Luo, L.S. Dns and les of decaying isotropic turbulence with and without frame rotation using lattice boltzmann method. *Journal of Computational Physics*, 209(2):(2005), 599–616.

[4] QIAN, Y.; DHUMIERES, D.; LALLEMAND, P. Lattice bgk models for navier-stokes equation. *EUROPHYSICS LETTERS*, 17(6BIS):(1992), 479–484.

[5] Xu, A.G.; Zhang, G.C.; Gan, Y.B.; Chen, F.; Yu, X.J. Lattice boltzmann modeling and simulation of compressible flows. *FRONTIERS OF PHYSICS*, 7(5):(2012), 582–600.

[6] Godenschwager, C.; Schornbaum, F.; Bauer, M.; Koestler, H.; Ruede, U. A framework for hybrid parallel flow simulations with a trillion cells in complex geometries. In *2013 INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS (SC)*, International Conference for High Performance Computing Networking Storage and Analysis. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE (2013). International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Denver, CO, NOV 17-22, 2013.

[7] Argentini, R.; Bakker, A.F.; Lowe, C.P. Efficiently using memory in lattice boltzmann simulations. *FUTURE GENERATION COMPUTER SYSTEMS*, 20(6):(2004), 973–980. International Conference on Computational Science (ICCS 2003), ST PETERSBURG, RUSSIA, JUN 02-04, 2003.

[8] Latt, J.; Chopard, B.; Malaspinas, O.; Deville, M.; Michler, A. Straight velocity boundaries in the lattice boltzmann method. *PHYSICAL REVIEW E*, 77(5, 2).

[9] Inamuro, T.; Yoshino, M.; Ogino, F. A non-slip boundary condition for lattice boltzmann simulations (vol 7, pg 2928, 1995). *PHYSICS OF FLUIDS*, 8(4):(1996), 1124.

[10] Zou, Q.S.; He, X.Y. On pressure and velocity boundary conditions for the lattice boltzmann bgk model. *PHYSICS OF FLUIDS*, 9(6):(1997), 1591–1598.

[11] Guo, Z.L.; Zheng, C.G.; Shi, B.C. An extrapolation method for boundary conditions in lattice boltzmann method. *PHYSICS OF FLUIDS*, 14(6):(2002), 2007–2010.

[12] SKORDOS, P. Initial and boundary-conditions for the lattice boltzmann method. *PHYSICAL REVIEW E*, 48(6):(1993), 4823–4842.

[13] Bouzidi, M.; Firdaouss, M.; Lallemand, P. Momentum transfer of a boltzmann-lattice fluid with boundaries. *PHYSICS OF FLUIDS*, 13(11):(2001), 3452–3459.

[14] Neumann, P.; Neckel, T. A dynamic mesh refinement technique for lattice boltzmann simulations on octree-like grids. *COMPUTATIONAL MECHANICS*, 51(2):(2013), 237–253.

[15] Weinzierl, T.; *et al.* Peano—a Framework for PDE Solvers on Spacetree Grids (2017). Http://www.peano-framework.org.

[16] Geier, M.; Greiner, A.; Korvink, G.J. Bubble functions for the lattice boltzmann method and their application to grid refinement. *The European Physical Journal Special Topics*, 171(1):(2009), 173–179.

[17] Rohde, M.; Kandhai, D.; Derksen, J.J.; van den Akker, H.E.A. A generic, mass conservative local grid refinement technique for lattice-boltzmann schemes. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 51(4):(2006), 439–468.

[18] Yu, D.Z.; Mei, R.W.; Shyy, W. A multi-block lattice boltzmann method for viscous fluid flows. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 39(2):(2002), 99–120.

[19] Filippova, O.; Hänel, D. Grid refinement for lattice-bgk models. *Journal of Computational Physics*, 147(1):(1998), 219–228.

[20] Chen, H.; Filippova, O.; Hoch, J.; Molvig, K.; Shock, R.; Teixeira, C.; Zhang, R. Grid refinement in lattice boltzmann methods based on volumetric formulation. *PHYSICA A-STATISTICAL MECHANICS AND ITS APPLICATIONS*, 362(1):(2006), 158–167. 13th International Conference on Discrete Simulation of Fluid Dyanmics, Cambridge, MA, AUG 16-20, 2004.

[21] Dupuis, A.; Chopard, B. Theory and applications of an alternative lattice boltzmann grid refinement algorithm. *Phys. Rev. E*, 67(6):(2003), 066707.

[22] Hou, S.; Zou, Q.; Chen, S.; Doolen, G.; Cogley, A. Simulation of cavity flow by the lattice boltzmann method. *JOURNAL OF COMPUTATIONAL PHYSICS*, 118(2):(1995), 329–347.

[23] Iwatsu, R.; Ishii, K.; Kawamura, T.; Kuwahara, K.; Hyun, J.M. Numerical simulation of three-dimensional flow structure in a driven cavity. *Fluid Dynamics Research*, 5(3):(1989), 173.

[24] Albensoeder, S.; Kuhlmann, H.C. Accurate three-dimensional lid-driven cavity flow. *Journal of Computational Physics*, 206(2):(2005), 536–558.

[25] Leriche, E.; Gavrilakis, S. Direct numerical simulation of the flow in a lid-driven cubical cavity. *Physics of Fluids (1994-present)*, 12(6):(2000), 1363–1376.

Chapter 4

Particle modelling

After achieve the stationary regime of the flow field the incoming particles start to go into the domain. To obtain a realistic and representative behaviour the set of particles has to replicate the typical populations measured during the working conditions of a Diesel engine. This populations cover particles whose diameter ranges from 30nm to several μm. The following sections describe how the population is, the calculation of the trajectories and the techniques used to obtain the parameters which govern their motion.

4.1 The Diesel particles

In Diesel emission context, a particle is every matter in liquid or solid state dilute into the gas in normal conditions. The first step to provide a realistic population of particles and consequently a good model is a good knowledge base about them. This section describes the formation process and the statistical distribution. Both of them are equally important to understand the properties to be used afterwards.

4.1.1 Formation process

The formation process helps us to understand some properties of the particles such as their shape or their density in a direct way, or the drag coefficient and the diffusion coefficient in a indirect one. This process is very complex to be described in detail at this document. In fact, it is still a leading edge topic nowadays. However a brief introduction gives the hint to understand the

distribution in size and the effect of the filter on it.

The formation process starts during the combustion. During the chemical reaction the hydrocarbon chains reacts producing principally CO_2 and water. However traces of other chemical products could appear such as CO, NO_x or partially burned hydrocarbons. The precursors of the Diesel particles are the last ones. During combustion some of the chains lost some hydrogen becoming into unsaturated chains with double and triple links. If the process continues the hydrogen vs carbon proportion is too small and the chain closes on itself creating cyclic compounds, like benzene and other aromatics. Every cyclic radical could find other radicals and combine. Consequently poly-cyclic compounds start to grow up, becoming larger. This compounds are called PAH, an abbreviation of Poly-cyclic Aromatic Hydrocarbons. In the limit this material lost practically all the hydrogen and a graphene film is obtained. When several films of graphene find each other van der Waals forces maintain them together in a stack of graphite. Again, each piece of graphite attracts each other. When several of these crystals gather together the resultant structure takes the shape of a sphere. The inner pieces are randomly oriented in contrast with the outer ones, whose normal points out the sphere. This outer layer creates a shell around the agglomerate. This structure is already considered as a particle. Because this type of particles are the master pieces to create the bigger ones, they are usually called precursor particles. When several particles collide, they can coalesce or they can aggregate. The coalescence of several particles results into a new particle. The aggregation put several particles together, but the shape of every particle is maintained. Its final shape is an aggregate of spheres. Figure 4.1 resumes the process.

4.1.2 Characterisation

There are no equal particles. Each one varies in size, composition and shape. These differences are specially important for large aggregates. At this point the geometry is chaotic and the parameters which resume their properties such as the drag coefficient present high uncertainty. Different measurements techniques are summarised in the review written by Sorensen [1]. In this review it is studied both geometrical properties such as fractal dimension and mobility radius, an indirect measurement of the aerodynamic properties. Exact measurements becomes very difficult and equivalent descriptors are used instead. The reference pattern used for the equivalence is the spherical particle whose density is 1000kg/m^3. From this definition it is very easy to calculate the equivalent aerodynamic diameter, probably the most important descriptor of this field. Because of the small size of the particles, they can be considered immersed into Stokes flow. The drag force F_D of an sphere of diameter D in

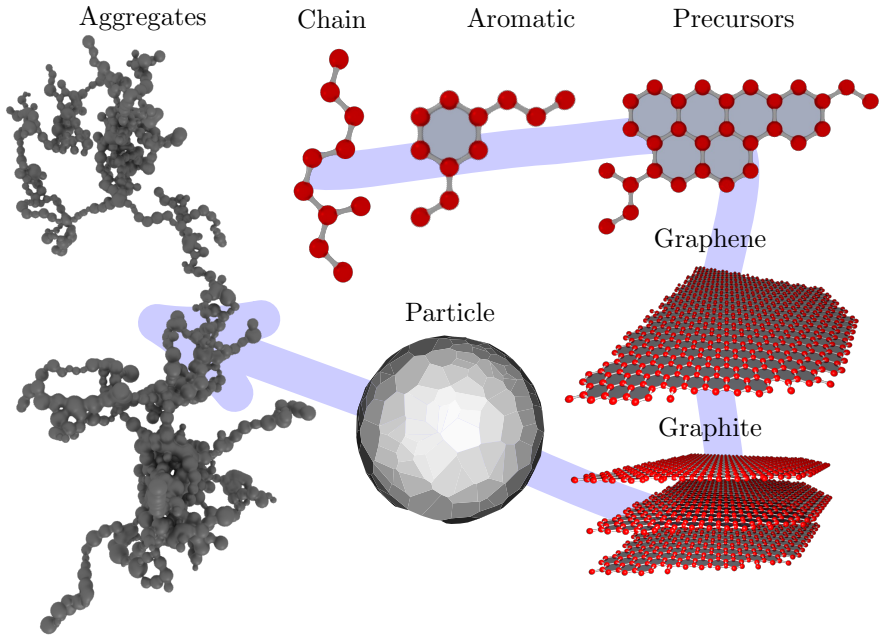

Figure 4.1: Formation process of Diesel particles.

Stokes regime is

$$F_D = 3\pi\mu D U_\infty \tag{4.1}$$

when it is immersed into a fluid with unperturbed relative velocity U_∞ and viscosity μ. If the same sphere is also affected by a constant external force like gravity, centrifugal force or electromagnetism (if the particle is electrically charged) it will travel at constant velocity after a while. At this point the forces will be in equilibrium, without the influence of the kinetic effects. If the particle moves due to the gravity force, taking into account its density, the total force is

$$g\rho_p \pi D^3/6 - 3\pi\mu D U_\infty = 0, \tag{4.2}$$

where g is the gravity force per unit mass and ρ_p is the density of the particle. The equivalent diameter can be obtain measuring its speed as

$$D_{eq} = \sqrt{\frac{18\mu U_\infty}{\rho_{eq} g}}. \tag{4.3}$$

Similar results can be obtained for other forces like centrifugal force or electromagnetic force. Accordingly to the aerodynamic diameter Diesel particles are classified as:

- PM_{10}: particles whose aerodynamic diameter is lower than 10μm.
- Fine particles: particles whose aerodynamic diameter is lower than 2.5μm.
- Ultrafine particles: particles whose aerodynamic diameter is lower than 0.1μm.
- Nanoparticles: particles whose aerodynamic diameter is lower than 0.05μm.

Once it is clear how the properties of the particles are measured it is time to analyse how those measurements are. One of the most interesting measures is the concentration of particles as a function of its aerodynamic diameter. The typical population can be seen at figure 4.2. It can be appreciated how the distribution is the sum of three overlapping log-normal curves, each one representing a different type of particles.

Nuclei mode: particles whose aerodynamic diameter is lower than 50nm. It contains between the 50% and the 90% of the particles.

Accumulation mode: the population of this mode has an aerodynamic diameter between 50nm and 1μm. It contains between the 80% and 95% of the mass.

Coarse mode: particles whose aerodynamic diameter is larger than 1μm. The number of particles contained by this mode is negligible. However, due to their size in comparison with other modes, the contained mass can be representative enough to take this distribution into account.

4.2 Particle transport

Capturing particles is the main purpose of every filter. This magnitude can not be quantified without knowing their trajectories. Consequently calculating correctly the trajectory becomes an important requirement of filter simulations. Depending on the concentration the particles can be introduced in the formulation as a dispersed phase to be solved with Eulerian formulation or discrete elements to be tracked by a Lagrangian solver. The first approximation is adequate when each discrete element (mathematical entity of the flow field to be solved numerically) contains a representative population of particles. The dispersed phase can be approximated as a continuum medium. When the number of particles per discrete element is very low continuum media is not valid anymore. Lagrangian formulation is necessary to track each particle separately. The case of interest is of the second kind. Trajectories of Diesel particles are computed using Lagrangian formulation.

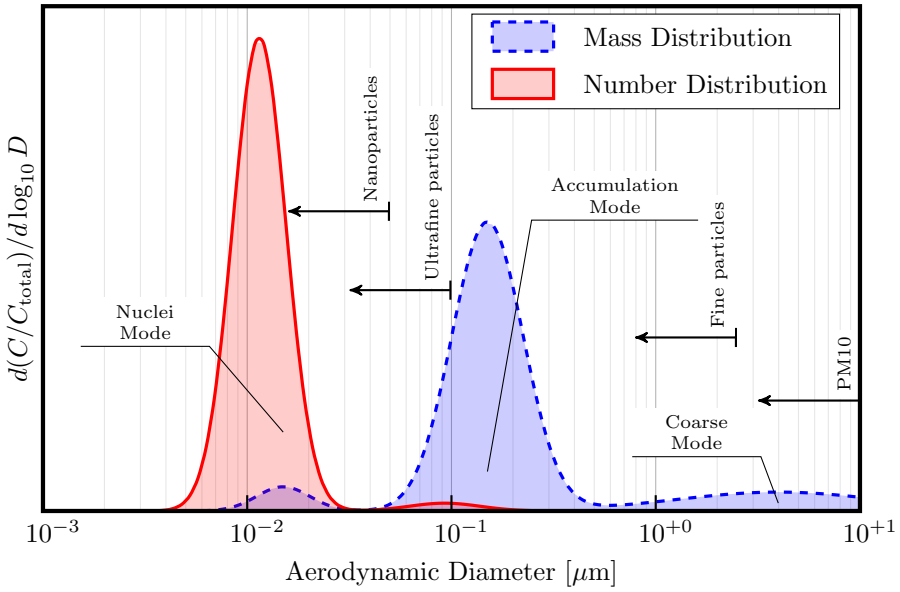

Figure 4.2: Typical numeric and mass distribution of a Diesel particle population respect to the aerodynamic diameter.

If the relative size of a particle with respect to lattice spacing is very low the particles can be considered zero-dimensional. Otherwise 3D effects have to be considered. Dietzel, Ernst and Sommerfeld [2] use both approximations to simulated deposition. The first part of their investigation concerns to coupling Lagrangian formulation of point-particles with LBM. The second part consists on calculations of agglomerates with detail. The aerodynamic forces of real geometries are calculated for several fractal dimensions, a magnitude which quantifies how compact is an agglomerate. The solver to realise this work is explained by Dietzel and Sommerfeld in a previous publication [3].

An important part of the calculation of filtering process is particle transport. The trajectory has to be calculated accurately to predict how many particles cross the filter. Capture the particles is the purpose of the filter and This magnitude defines As it has been seen in the previous section the particle size to be included in the simulation goes from aerodynamics diameters of some tens of nm to several μm. Several physical phenomena affect the particle motion, such as Brownian motion, inertial forces, continuity of the media among others. The dominant effects depend on the particle size. Some of them, like Brownian motion, only affects the trajectory of small particles. On the contrary, inertia is only important for large particles. The following effects have been considered:

- Trajectory calculation technique. The number of particles to be calculated simultaneously in the domain is relatively low. Consequently each particle has to be simulated independently instead of use the statistical description of its population. The Lagrangian formulation solves better this type of problems. See section 4.2.1 for more details.

- Drag Force. It is considered Stokes regime around the particle. The validity range of this hypothesis is explained in detail at section 4.2.2. The most important parameter related with drag force is particle size. Fluid mechanics of rarefied flows is slightly different to continuum media. The drag force of small particles, immersed in this type of fluids, has a deviation with respect to Stokes flow for continuum media. The deviation is modelled by a correction factor.

- Particle-Fluid interaction. When the drag force slows down the particle then the particle also pushes the surrounding fluid. Depending on the particle size the drag force could be negligible or not. Furthermore, if the particle is very large, the aerodynamic effects on the surrounding air could be extended along several lattice points. In fact, due to the complexity of the geometry, each agglomerate has to be considered as a porous particle, so some flux crosses it. How the particle interacts with the surrounding fluid is explained in section 4.2.3

- Brownian motion. The particles diffuse in the air. The counterpart in discrete formulation of the Fick's law is the Brownian motion. Because the particle size differs a lot the Brownian motion can be extremely strong for nanoparticles, and it can be negligible when the aerodynamic diameter is several μm long. The section 4.2.4 explains the ranges where Brownian motion has to be taken into account and the techniques used to calculate it depending on the particle size.

4.2.1 Trajectory calculation

There are two ways for solving differential equations. In one hand, the Eulerian formulation. A control volume is fixed and the differential equation is solved inside considering the net flux across its limits. In the other hand is the Lagrangian formulation. Instead of counting how many particles goes in and out the control volume, each particle is calculated separately, following its trajectory and the interactions with the surrounding particles. In general way the first formulation is appropriate when the number of particles inside the volume is large enough to be statistically representative. Otherwise continuum solutions are achieved when the real phenomenon is discrete. The second formulation generally works better when the number of particles is very low and the physics to reproduce with the simulation is discrete.

Typical concentrations before the Diesel Particulate Filter (DPF) are around 10^8 part./cm^3. The order of magnitude of the domain size is 1mm^3. As result, the upper threshold for the number of particles in the domain is around 10^5 particles, a reachable enough value for a computer if each particle is calculated separately. Another interesting value related with the concentration is the number of particles per lattice node. Each node contains the information of the surrounding volume, approximately 1μm. It means that each node contains 10^{-4} particles on average. Only one cell in every ten thousand contains the centre of a particle, reason why the particles have to be considered as a discrete field. As resume, the total number of particles to be simulated is clearly reachable. They can be calculated one by one. Their low concentration per cell bans the use of statistics. As conclusion the most appropriate formulation is the Lagrangian.

The next step is to specify the equation which describes the particle motion. Supposing it as zero-dimensional, only the three translational degrees of freedom have to be considered, resulting in the three-component vector expression

$$m_p \ddot{\vec{x}}_p = \vec{F}_D + \vec{F}_B + \vec{F}_{g+b}, \tag{4.4}$$

where m_p is the particle mass, $\ddot{\vec{x}}_p$ is the second derivative of the location of the particle with respect to time (acceleration) and $\vec{F}_D$, $\vec{F}_B$ and $\vec{F}_{g+b}$ are the drag force, Brownian force and gravity together with buoyancy forces respectively. This equation of second order can be decomposed in two of first order introducing the particle speed $\vec{u}_p$ as a variable

$$m_p \dot{\vec{u}}_p = \vec{F}_D + \vec{F}_B + \vec{F}_{g+b}, \quad \vec{u}_p = \dot{\vec{x}}_p. \tag{4.5}$$

Solving for the first derivative is immediate. The Runge-Kutta method is used for integration. The LBM is a second order method as well as the interpolation, which is a second order too. Taking this into account the best option is to use a second order time integrator, which is as accurate as the interpolator and the method and does not increase the complexity and the resources consumption required by higher orders. Defining the vector of independent variables $\vec{q} = \{\vec{x}, \vec{u}_p\}$ and its derivative in time as $\dot{\vec{q}} = \{\dot{\vec{x}}, \dot{\vec{u}}_p\}$ it is very easy to set out the equality $\dot{\vec{q}} = \vec{f}(\vec{q})$ from expression 4.5. The first approximation to the derivative is $\vec{\kappa}_0 = \vec{f}(\vec{q})$. The second approximation is quite similar, $\vec{\kappa}_1 = \vec{f}(\vec{q} + \Delta t \vec{\kappa}_0)$. The next step can be easily approximated as $\vec{q}(t + \Delta t) = \vec{q}(t) + \Delta t(\vec{\kappa}_0 + \vec{\kappa}_1)/2$.

4.2.2 Drag force

The aerodynamic drag is the projection of the resultant force on the opposite direction to the relative motion between an immersed body and the surroun-

ding fluid due to their difference in velocity. It is usually very difficult calculating it algebraically, reason why a semi-empirical non-dimensional coefficient is usually employed. This coefficient, by definition, is the result of dividing the total drag force F_D between the dynamic pressure $q = \rho\|\vec{u}_r\|^2/2$ ($\vec{u}_r$ is the relative speed defined as the difference $\vec{u}_f - \vec{u}_p$) and a reference area S. It is called aerodynamic drag coefficient C_D

$$F_D = C_D \frac{1}{2}\rho\|\vec{u}_r\|^2 S. \tag{4.6}$$

Recovering the notation and writing it in vector form the equations becomes

$$\vec{F}_D = \frac{1}{2} S C_D \rho_f (\vec{u}_r \cdot \vec{u}_r) \frac{\vec{u}_r}{\|\vec{u}_r\|} \tag{4.7}$$

But the aerodynamic drag coefficient can also depends on the velocity and the properties of the fluid. Some hypothesis can be applied to simplify the expressions of the force or to make negligible some of them. The range of validity of an hypothesis is called regime. Two sets of hypothesis delimits the regimes of this context. The first one is related with inertial and viscous effects. The indicator is the Reynolds number Re. The second set depends on the ratio between the characteristic scale of the immersed body and the separation between the molecules of the fluid. The indicator is the Knudsen number Kn. Both indicators are dimensionless numbers and their use is very extended in fluid dynamics. The Reynolds number, defined as

$$Re_L = \frac{\rho U_\infty L}{\mu}, \tag{4.8}$$

it is strongly related with the chaos in the flow field and the fluid patterns of the flow. Very low Reynolds means high viscosity effects in comparison with the kinetic ones. Any perturbation is damped and the flow practically follows the wall of the immersed objects without detachment. This regime is called Stokes regime and it is in fact the case of the Diesel particles. Other regimes are the laminar regime, when the Reynolds number is moderate, and the turbulent regime, when the Reynolds number is very high. In laminar flows there are some detachments in regions with high curvature, but the resultant structures are coherent and predictable. In turbulent flows the kinetic energy of the fluid is very high and any perturbation could be amplified and produce chaotic structures.

The other interesting regime indicator for the study of Diesel particles is the Knudsen number. It is defined as the quotient between the mean free path λ, the mean distance between two consecutive collisions of a molecule with others of the same fluid, and the characteristic scale of the problem

$$Kn = \lambda/L. \tag{4.9}$$

It represents the limit where the continuum hypothesis for the flow field is not valid anymore. High Knudsen number means very low concentration of particles so the space between collisions is much longer than the immersed object. It happens when the pressure is very low or when the object is very small. This regime is called high vacuum regime or free molecular regime. For moderate Knudsen number the regime receives the same name than the dimensionless number, Knudsen regime. The concentration of molecules is high but it is not high enough to achieve the equilibrium distribution, so the NS equation are not valid yet. However LBM can calculate this regime. The last regime, when the mean free path is much lower than the characteristic scale, is the continuous regime. The smallest Diesel particles are several tens nm thick. The mean free path is the same order of magnitude long. As consequence, to calculate the drag coefficient of the particles correctly, the effects of the Knudsen number have to be taken into account.

Now it is time to specify the drag coefficients for the case of interest. The equivalent spherical Diesel particle (see section 4.1.2) is moving into a fluid at Stokes regime and in continuum or Knudsen regime depending on its diameter. The motion equation for an spherical particle immersed in Stokes flow is

$$m_p \vec{\dot{u}}_p = 3\pi\mu C_c d_p (\vec{u}_f - \vec{u}_p) \tag{4.10}$$

where C_c is a constant depending on the Knudsen number to correct the non-continuum effects. The premultiplicative terms of the right hand side of the previous expression can be put together to form a time constant

$$\tau_d = \frac{m_p}{3\pi\mu C_c d_p} \tag{4.11}$$

getting

$$\frac{\partial u_p}{\partial t} = \frac{t}{\tau_d} (u_f - u_p) . \tag{4.12}$$

The velocity of the particle for a uniform velocity flow field is obtained by solving the differential equation

$$u_p(t) = u_f + (u_p(0) - u_f) \exp(-t/\tau_d) . \tag{4.13}$$

How fast the particle gets the fluid velocity is dictated by the time constant τ_d. Large values of τ_d means large inertia and consequently the particle needs a lot of time to achieve the velocity of the fluid. In the limit, if τ_d is extremely large, it can be considered the movement of the particle as independent of the flow field. This case does not appear in the Diesel particle context. On the contrary, low values of τ_d are representative of viscous-dominant motion. The particle follows the flow field. Its trajectory can be approximated accurately by a pathline. For time non-dependent flow fields and in the absence of other forces the trajectory of this type of particles can be calculated during the post-processing stage.

4.2.3 Particle-Fluid interaction

If the flow slows down the particle, the particle reacts pushing the fluid. It is a direct consequence of the Newton's third law. The particle motion and the flow field is then intrinsically coupled. Coupling the particle movement with the flow field solver slows down the calculation, so it is very interesting to see when decoupling has no consequence in the accuracy of the solution. There are two situations when the particle-fluid interaction can be considered negligible:

- The particle is so small that its characteristic linear momentum is insignificant compared with the linear momentum concentrated in a node of the lattice, $m_p U_p \ll \rho_f h^3 U_f$. The presence of particles does not modifies significantly the local flow field.
- The viscous effects are so strong in comparison with the kinetic and external forces that the particle is forced to follow the pathline. In this situation the particle gets immediately the velocity of the surrounding flow. Consequently, because the drag force depends of the relative velocity of the surrounding fluid with respect to the immersed object and this is zero, the drag force is also zero.

For the rest of cases decoupling can affect both the trajectories and the flow field, specially if the particles are large and they are immersed in high gradient regions. Depending on the size of the particle, this force can affect a single node or a set of them. When the perturbation of the particle affects multiple nodes all of them contributes to the drag force. The reaction force is also shared among the affected area.

How the force is distributed along the surrounding nodes is dictated by a Kernel function $W(\vec{x}, \vec{\xi})$. $\vec{x}$ is the location of a generic point of the domain and $\vec{\xi}$ is the location of the centre of mass of the particle which perturbs the fluid. The objective of the Kernel is weighting the contribution of each node to the total drag of the particle as well as the reaction forces.

One of the most typical Kernels is the Gaussian Kernel

$$K(\vec{x}, \vec{\xi}) = (2\pi\sigma)^{-D/2} \exp\left(-\frac{\|\vec{\xi} - \vec{x}\|^2}{2\sigma^2}\right) \tag{4.14}$$

where D is the number of dimensions. The Kernel is in fact the Gaussian distribution. However, instead of calculating the contribution of every cell by integration, the Kernel is evaluated for every location of the surrounding nodes $K(\vec{x}_i, \vec{\xi})$. It gives to each node a different weight depending on the distance to the centre of the particle $\vec{\xi}$. The total contribution of all the weights is

not normalised to one. Then each evaluation is divided by the sum of all evaluations. It returns the weight for the i-th node of the lattice

$$w_i = K(\vec{x}_i, \vec{\xi}) \Big/ \sum_i K(\vec{x}_i, \vec{\xi}) \ , \qquad \text{with} \qquad \sum_i w_i = 1. \tag{4.15}$$

The computational cost of the Kernel can be reduced deleting the premultiplicative constant

$$K(\vec{x}, \vec{\xi}) = \exp\left(-\frac{\|\vec{\xi} - \vec{x}\|^2}{2\sigma^2} \right). \tag{4.16}$$

However its cost is still very expensive for two reasons. Firstly the exponential function needs a long time to be evaluated in comparison with others such as multiplication or addition. Secondly the function returns values different to zero everywhere. Truncating the function for larger distances from the particle solves the second problem. However a better solution is to replace the exponential by a piecewise-defined polynomial.

A novel efficient Kernel is proposed for performance improvement. The Kernel approximates the Gaussian distribution. It have several peculiarities:

- It is symmetrical respect to 0.
- It is continuous in value and first derivative.
- The polynomial depends on even powers of the distance. It saves computational cost avoiding the use of the root square.

Three trams have been considered to approximate the Gaussian distribution. The last one is zero constant. Considering $z = \|\vec{x} - \vec{\xi}\|/\sigma$, the first and second ranges take the form $y = a + bz^2 + cz^4$. The generic equation is then

$$y(z) = \begin{cases} a_0 + b_0 z^2 + c_0 z^4 & \forall x \in [0, z_0) \\ a_1 + b_1 z^2 + c_1 z^4 & \forall x \in [z_0, z_1) \\ 0 & \forall z \in [z_1, \infty) \end{cases} \tag{4.17}$$

There are eight unknowns $\{a_0, b_0, c_0, a_1, b_1, c_1, x_0, x_1\}$. The values of these constants have to minimise the quadratic error with respect to the Gaussian distribution. The function to be minimised is

$$E^2 = \int_0^\infty \left(y(z) - \exp(z^2/2) \right)^2 \, dz. \tag{4.18}$$

The constrains are continuity in value and first derivative

$$a_0 + b_0 z_0^2 + c_0 z_0^4 = a_1 + b_1 z_0^2 + c_1 z_0^4 \qquad \text{Continuity at } z_0, \quad (4.19a)$$

$$a_1 + b_1 z_1^2 + c_1 z_1^4 = 0 \qquad \text{Continuity at } z_1, \quad (4.19b)$$

$$2b_0 z_0 + 4c_0 z_0^3 = 2b_1 z_0 + 4c_1 z_0^3 \qquad \text{Derivative continuity at } z_0, \quad (4.19c)$$

$$2b_1 z_1 + 4c_1 z_1^3 = 0 \qquad \text{Derivative continuity at } z_1. \quad (4.19d)$$

Applying the Lagrange multipliers technique the unknowns are solved. The new Kernel approximates the Gaussian distribution with minimum quadratic error

$$K(z) = \begin{cases} 1.0000 - 0.4517z^2 + 0.064379z^4 & \forall z \in [0,\ 1.6895), \\ 0.5367 - 0.1271z^2 + 0.007523z^4 & \forall z \in [1.6895,\ 2.9064), \\ 0 & \forall z \in [2.9064,\ \infty). \end{cases} \quad (4.20)$$

Figure 4.3 shows both the exponential and the polynomial approximation as well as the difference between them. The Kernel models the distribution of the

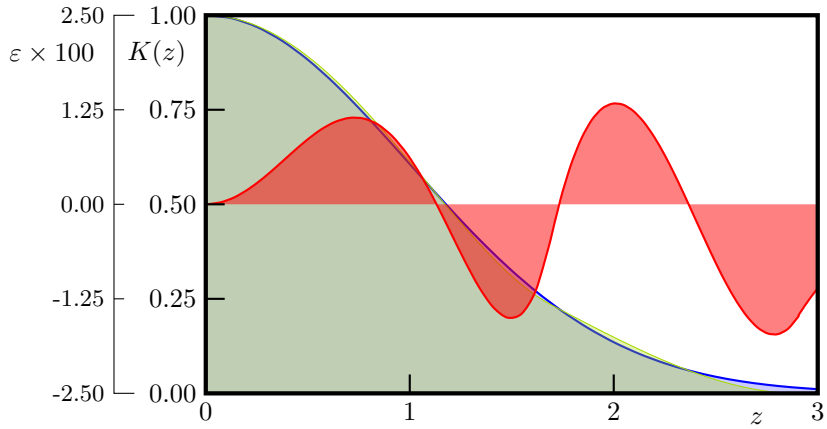

Figure 4.3: (blue) Gauss Kernel, (green) polynomial approximation and (red) error.

drag force in space. Instead of being concentrated in a single point, the particle interacts with an extended region. The intensity which the particle interacts with depends on the distance. The total drag force has to be identical to the drag force produced by the equivalent sphere in a homogeneous medium

$$\vec{\mathscr{F}}_D = \sum_i 3 w_i \pi \mu C_C d_p (\vec{u}_i - \vec{u}_p) \quad (4.21)$$

where w_i is calculated for every node of the surrounding lattice. The reaction force for each of these nodes is the corresponding component of the sum but with opposite direction.

4.2.4 Brownian motion

Brownian motion is the random movement of immersed particles produced by impacts with the molecules of the surrounding fluid. It is the discrete counterpart of the Fisk's law

$$\frac{\partial \phi}{\partial t} = \mathscr{D}\frac{\partial^2 \phi}{\partial x^2}. \tag{4.22}$$

The Green's function for the previous equation is

$$G(x,t;\xi,\tau) = (4\pi\mathscr{D}(t-\tau))^{-3/2}\exp\left(-\frac{(x-\xi)^2}{4\mathscr{D}(t-\tau)}\right). \tag{4.23}$$

This equation is the solution of continuous media after the diffusion of a single spot located at $\vec{\xi}$ and time τ. An equivalent distribution has to be obtained after computing the location of multiple particles using Lagrangian formulation 4.4. In other words, the normalised Green function is the probabilistic field for the location of a single particle $\vec{x}$ at time t originally located at $\vec{\xi}$ at time τ. The next position of the particle can be modelled by random samples extracted from this probabilistic density distribution. It is, in fact, a Gaussian distribution centred in the expected location with standard deviation $\sigma = \sqrt{2\mathscr{D}\Delta t}$. This formulation was originally introduced by Einstein in 1926 [4].

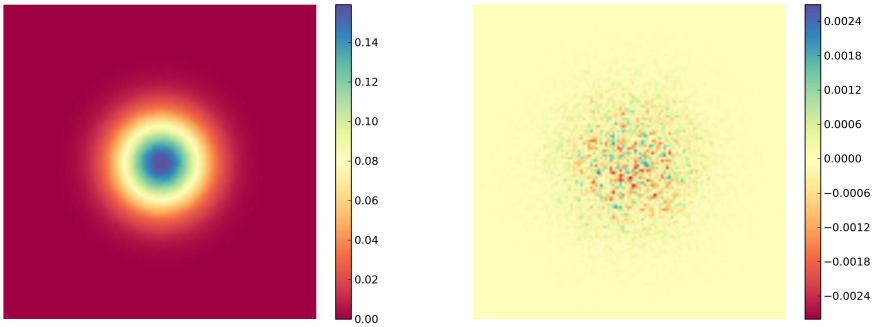

Figure 4.4: (Left) Particle density concentration after simulating Brownian motion of 1M particles from the same origin after 1000 iterations. (Right) Difference between the result and the Green's function for the Fick's law.

The diffusion coefficient of the particles in the fluid $\mathscr{D}$ can be estimated from the drag coefficient and Knudsen number

$$\mathscr{D} = \frac{k_B T}{C_D} C_C. \tag{4.24}$$

The drag coefficient for a sphere is $C_D = 3\pi\mu d_p$ in Stokes regime.

Temporal scale disparity. Multi-step Brownian motion

The Brownian motion can be very important for very small particles. When the Brownian displacement for a lattice time step Δt is several times bigger than the lattice spacing the final location can not be directly calculated. The time step has to be refined to achieve accurate results. The covered distance by a single particle will be less than $3\sigma = 3\sqrt{2\mathscr{D}\Delta t}$ with a 96% of probability. This distance has to be less or equal to the lattice spacing. If it is not, the time step is divided in multiple sub-steps

$$N_{\mathrm{Br}} = \left\lceil \frac{h}{3\sqrt{2\mathscr{D}\Delta t}} \right\rceil . \tag{4.25}$$

The trajectory of the particle is completely refreshed for every sub-step, including drag force and other effects. The new random generator for Brownian deviation has a standard deviation of $\sigma = \sqrt{2\mathscr{D}\Delta t/N_{\mathrm{Br}}}$.

4.3 Computed generated particles. Ballistic collision.

The Diesel particles have been described by means of their equivalent aerodynamic diameter. This magnitude summarises the aerodynamic behaviour of the entire population correctly. However in addition to the aerodynamics there are other effects to take into account. Brownian motion is an important term for the smallest particles. 3D effects have to be considered for largest particles. Consequently the real distribution is needed. The aerodynamic equivalency is not valid in this context.

The Diesel particles can be approximated as agglomerates of spheres. The agglomeration process is simulated by a new solver developed *ad hoc* for this purpose. The relation between the real geometry and the aerodynamic equivalence can be found by studying the aerodynamics of the resulting agglomerates. Several assumptions have been considered to create the population. The first and probably the most important hypothesis is free regime. There is no fluid to interact with the particles. They moves freely accordingly to Newton's laws. Other important assumption is the shape of the precursors. They are supposed to be perfect spheres. This hypothesis approximates very well the reality.

Understanding the particle agglomeration process is of great interest for industry. Diesel soot is formed by agglomerates of particles indeed. But also nano-particles synthesis is affected by this phenomenon. Wang and Sorensen [1, 5] measure the gyration radius of aggregates and their diffusion coefficient

by light scattering. The measurements are used to calculate the fractal dimension for several Knudsen numbers. Similar results are obtained from simulations. Eggersdorfer and Pratsinis [6] study the morphology of agglomerates generated from polydisperse particles. They generate agglomerates employing multiple algorithms. The objective is to quantify the effects of the precursor size into the final geometry. They extend their work [7] to study the agglomeration process for continuum, rarefied and free molecular regimes. Their work is a complete study about agglomeration and also includes aggregation and sintering. This study is extended for larger agglomerates by Goudeli, Eggersdorfer and Pratsinis [8]. The investigation continues with evolution in time [9]. Morgan *et al.* [10] simulate generation of Diesel particles. The geometry of resultant populations is statistically studied. They correlate different geometrical descriptors such as fractal dimension, projected area, number of precursors, *etc.* Dekkers and Friedlander [11] generates the agglomerates by Brownian motion of the precursors. They used the results to find the relation between the fractal dimension with the Knudsen number.

In the following lines it is explained how the solver calculates the agglomerate motion and formation.

4.3.1 Theoretical background

Particle generation

The agglomerates are chaotic and the chaos begins with the size of the primary particles. Because each precursor is considered as a sphere only the radius is needed to describe its geometry. The population of precursors follows a log-normal distribution in size.

Any population of particles with different mass and velocity gets the equilibrium distribution with time. The equilibrium distribution is the Maxwell-Boltzmann distribution. The Maxwell-Boltzmann distribution relates the velocity distribution with the mass of the particles. Concretely the standard deviation in velocity is related with the particle mass $\mathscr{M}$. If σ_1 is the standard deviation in velocity for particles with unitary mass, the generic standard deviation is $\sigma = \sigma_1/\sqrt{\mathscr{M}}$. A particle four times heavier than the reference is twice slower. The Maxwell distribution is the combination of three Gaussian distributions, one for each component of the speed. The probabilistic density distribution is

$$\mathrm{pdf}(\vec{c}) = \left(\frac{2\pi\sigma_1^2}{\mathscr{M}}\right)^{-D/2} \exp\left(-\frac{\|\vec{c}-\vec{u}\|^2}{2\sigma_1^2/\mathscr{M}}\right), \tag{4.26}$$

where $\vec{u}$ is the mean velocity. For ideal gases $\sigma_1 = \sqrt{k_B T}$. All of these is

for equilibrium. However, for the sake of simplicity, the same distribution in speed has been considered. If it is wanted, the effect of the particle mass can be taken into account after generating the velocity population. It suffices with scaling the velocity by the factor coming from the previously generated mass respect to the mean value

$$\vec{c}^{*} = (\vec{c}_0 - \overline{\vec{u}})\sqrt{\mathscr{M}_{\text{ref}}/\mathscr{M}} + \overline{\vec{u}}. \tag{4.27}$$

Independently of the initial distribution, the population tends to equilibrium with time. For long enough simulations the solution will be independent of the initial conditions.

The particles can also rotate. To simplify the calculus each single particle has been created without rotation. However initial distributions can be considered if it is wanted. The methodology is similar to create the population of linear speeds. Again, if in addition the uniqueness of each particle is considered the rotational speed has to be corrected, this time by the effect of the moment of inertia.

Initial condition. Distribution in space.

In the previous section it is explained how the particles are created. Now it is time to described how the created particles are located. The simplest case is the cuboid domain. The location can be directly specified by three random samples of uniform distributions.

However the box is not the generic case. If the domain is discretised by tetrahedrons then each one has to contain the appropriate number of particles. Supposing n as the number of particles per unit volume, the particles in the i-th volume is $N_i = n \cdot V_i$. The result is not usually an integer. Taking the floor of this value as the number of particles to be created inside the i-th volume underestimates the total number of particles. Just as an example, if the concentrations of particles is 999 per unit volume and the unitary volume is divided by 500 equal elements then each element will contain only 1 particle returning a total of 500 particles when 999 are expected. To solve this problem a random sample χ coming from a uniform distribution and bounded by the interval $[0, 1]$ is added before taking the integer part

$$N_i = \lfloor n \cdot V_i + \chi \rfloor . \tag{4.28}$$

With this procedure the final number of particles in the total volume will be closer to the expected value.[1]

[1]The spatial distribution of particles for low values of N_i is more uniform, preventing clustering and empty spaces, what in statistical terminology is called "reducing the occurrence".

Once the number of particles in each element is known it is time to locate them inside. The first step is to select one of the vertexes $\vec{v}_0$ of the tetrahedron. Then three edges are selected following a path. The four vertexes are linked by the vectors $\vec{e}_0 = \vec{v}_1 - \vec{v}_0$, $\vec{e}_1 = \vec{v}_2 - \vec{v}_1$ and $\vec{e}_2 = \vec{v}_3 - \vec{v}_2$. To locate the particle in the tetrahedron three random samples from uniform distribution χ_0, χ_1 and χ_2 are needed. Calculating the final location requires knowing the values of three scaling factors

$$\vec{x} = s_0\vec{e}_0 + s_1\vec{e}_1 + s_2\vec{e}_2. \tag{4.29}$$

Two different things have to be taken in consideration. In first place the scale factor s_1 depends on the previous scale factor s_0, and the last scale factor s_2 depends on both s_0 and s_1. Secondly the probability to obtain a specific value is not uniform for the scale factors s_1 and s_2. The procedure to compute them is as follows. The first random sample fixes a section of the tetrahedron parallel to the opposite face of the first vertex. Because the section scales linearly from 0 to the last area the Probability Density Function (PDF) has to scale accordingly. A transformation of χ_0 to χ_0^* is needed. Considering that both distributions have to be equivalent

$$\int_0^{\chi_0} d\chi = \int_0^{\chi_0^*} 2\chi^*\, d\chi^*. \tag{4.30}$$

Solving the integrals returns how $\chi_0^* = \sqrt{\chi_0}$. With this simple operation the first scale factor is obtained $s_0 = \sqrt{\chi_0}$. The procedure to calculate the second factor is similar. The only difference is that the section has been already scaled by the previous scale factor, so $s_1 = s_0\sqrt{\chi_1} = \sqrt{\chi_0\chi_1}$. This procedure is also valid for the hyperspace if the i-th component is not the last one $s_i = \prod_i \sqrt{(\chi_i)}$. The last sample is equally probable, so it is not required the intermediate transformation to χ_i^*. That is, the last scale factor is $s_{D-1} = \chi_{D-1}\prod_{i=0}^{D-2}\sqrt{\chi_i}$. For the 3D case the factors are

$$s_0 = \sqrt{\chi_0}; \qquad s_1 = s_0\sqrt{\chi_1}; \qquad s_2 = s_1\chi_2. \tag{4.31}$$

Once the scaling factors have been calculated the location can be easily computed from expression 4.29.

Inlet boundary condition: reservoir of particles

When there is a reservoir next to the domain a net incoming flux appears from it. The incoming flux of particles with speed $\vec{c}$ per unit area and unit time injected by the reservoir can be easily calculated as

$$\phi_{N_{\vec{c}}} = n\,\mathrm{pdf}(\vec{c})\,\vec{c}\cdot d\vec{A}\,\delta t \qquad \forall \vec{c}\cdot d\vec{A} > 0. \tag{4.32}$$

Two conclusion can be obtained from expression 4.32. Firstly, the Probability Density Function of the tangent components does not change. Secondly, the number of incoming particles and consequently their speed is fixed by the expression 4.32. It modifies the original distribution multiplying by the velocity and rejecting the outgoing particles. Considering the Gaussian distribution as PDF the equation 4.32 becomes for the normal speed $c_n = \vec{c} \cdot \vec{n}$, where $\vec{n}$ is the normal to the wall $\vec{n} = \delta\vec{A}/\|\delta\vec{A}\|$,

$$\phi_N(\vec{c}_n) = \frac{n}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(c_n - c_{n,0})^2}{2\sigma^2}\right) c_n \, \delta t \delta A. \tag{4.33}$$

The uniform distribution has to be converted into a new one proportional to the flux given by equation 4.33. As every Cumulative Distribution Function (CDF), the values are limited by the interval [0,1], so the first step is to find the constant $1/K$ which forces the integral

$$\frac{1}{K} \int_0^\infty \exp\left(-\frac{(c_n - c_{n,0})^2}{2\sigma^2}\right) c_n \, dc_n = 1. \tag{4.34}$$

This integral is valid for every sample so it has to be calculated only once. The value of the constant is

$$K = \sqrt{\frac{\pi}{2}}\sigma c_{n,0}\left[\operatorname{erf}\left(\frac{c_{n,0}}{\sqrt{2}\sigma}\right) + 1\right] + \sigma^2 \exp\left(-\frac{c_{n,0}^2}{2\sigma^2}\right). \tag{4.35}$$

The number of incoming particles ΔN during a time step Δt in a section ΔA is obtained by the integration of equation 4.33 along the entire population of normal speeds c_n. It can be easily observed how the integral of the c_n-dependent terms is K from equation 4.34, so the total flux of particles which cross an area ΔA during a time step Δt is

$$\Delta N = \frac{n}{\sqrt{2\pi}\sigma} K \, \Delta t \Delta A. \tag{4.36}$$

Again, to take in consideration the decimal part, a random sample from a uniform distribution χ is added. After this step the integer part can be extracted

$$\Delta N = \left\lfloor \frac{n}{\sqrt{2\pi}\sigma} K \, \Delta t \Delta A + \chi \right\rfloor. \tag{4.37}$$

This operation is done each time that the number of incoming particles is calculated. The expression that converts a sample from the uniform distribution χ_i into a sample of normal velocity $c_{n,i}$ is

$$\frac{1}{K} \int_0^{c_{n,i}} \exp\left(-\frac{(c_n - c_{n,0})^2}{2\sigma^2}\right) c_n \, dc_n = \int_0^{\chi_i} d\chi. \tag{4.38}$$

Solving again this equation it is obtained the relation which returns the equivalence

$$\sqrt{\frac{\pi}{2}}\sigma c_{n,0}\left[\mathrm{erf}\left(\frac{c_{n,i}-c_{n,0}}{\sqrt{2}\sigma}\right)-1\right]-\sigma^2\exp\left(-\frac{(c_{n,i}-c_{n,0})^2}{2\sigma^2}\right)=K(\chi_i-1). \tag{4.39}$$

Solving for $c_{n,i}$ algebraically is not possible. An iterative method such as bisection or Newton-Raphson has to be applied. When a new particle coming from the reservoir goes into the domain the tangent components to the wall are two samples of the Gaussian distribution. The normal component is obtained generating a random sample from a uniform distribution χ_i and replacing and solving equation 4.39. Once the speed of the particle is known, it is time to set its location. The origin of the particle is somewhere in the triangle-shaped area ΔA. The exact location of the origin can be generated randomly applying equation 4.29. Firstly two edges are calculated from the coordinates of the vertexes $\vec{e}_0=\vec{v}_1-\vec{v}_0$ and $\vec{e}_1=\vec{v}_2-\vec{v}_1$. The final location is the sum of this two vectors scaled by the corresponding scale factors

$$\vec{x}_0=s_0\vec{e}_0+s_1\vec{e}_1. \tag{4.40}$$

The scales factors are obtained from two random samples χ_0 and χ_1 extracted from a uniform distribution bounded by the interval [0,1]. Uniform probability is obtained when the scales factors are $s_0=\sqrt{\chi_0}$ and $s_1=s_0\chi_1$. Now the point where the particle goes into the domain is known. But it covers a distance during a time shorter than the time step. The probability to go into the domain is not time dependent, it is uniform. As consequence this time can be generated randomly by the multiplication of the time step by a sample extracted from a uniform distribution. Consequently the particle is $\chi\Delta t$ old. During this time the particle covers a distance of $\vec{c}\Delta t\chi$. The final location is then

$$\vec{x}=\vec{x}_0+\chi\Delta t\vec{c}. \tag{4.41}$$

Mechanical properties of agglomerates

Each agglomerate is considered as a rigid body without restrictions. Six Degrees of Freedom (DoF) are enough to explain the motion of this type of bodies. Three of them specify the location and other three specify the orientation. The way a rigid solid interacts with forces and moments is governed by its properties, mass and inertial matrix. In this context the precursor is the simplest sample. It is just an spherical particle. Considering the particle of radius r made of uniform material whose density is ρ, the mass m and the inertial matrix respect to its centre of gravity $[I^0]$ are

$$m=\frac{3}{4}\pi\rho r^3,\qquad I^0_{ij}=\frac{2}{5}mr^2\delta_{ij}. \tag{4.42}$$

Here δ_{ij} is the Kronecker's delta[2]. When the centre of mass is not coincident with the origin of coordinates the inertial matrix $[I]$ becomes

$$I_{ij} = I^0_{ij} + m\left(d_0^2 + d_1^2 + d_2^2\right)\delta_{ij} - m x_i x_j. \tag{4.43}$$

Where $\vec{d}$ is the distance from the origin of coordinates to the centre of gravity. An agglomerate is composed by several of these precursors. The addition of the inertial properties of each precursor returns the inertial properties of the agglomerate. The first step is to calculate the total mass as the sum of the masses of its N precursors

$$m^{ag} = \sum_{n=1}^{N} m_n. \tag{4.44}$$

It is wanted to calculate the inertia matrix respect to the centre of mass

$$\vec{x}^{ag}_{cm} = \frac{1}{m^{ag}} \sum_{n=1}^{N} m_n \vec{x}_{cm,n}. \tag{4.45}$$

Finally the inertial matrix $[I^0]$ is obtained after adding all the contributions

$$I^{0\,ag}_{ij} = \sum_{n=1}^{N} \left[\frac{2}{5} m_n r_n^2 + m_n\left({d_0}_n^2 + {d_1}_n^2 + {d_2}_n^2\right)\right]\delta_{ij} - m_n x_{in} x_{j\,n}. \tag{4.46}$$

where $\vec{d}$ is the relative location of the centre of the precursor respect to the centre of mass of the agglomerate. Both the total mass m and the inertial matrix $[I]$ are used to calculate two magnitudes of interest, the angular and linear momentum $\vec{p}$ and $\vec{L}$, defined as

$$\vec{p} = m\vec{u}; \qquad \vec{L} = [I]\vec{\omega}, \tag{4.47}$$

where $\vec{u}$ is the linear velocity and $\vec{\omega}$ the angular velocity. Consequence of the third Newton's law, both the linear and angular momentum are conservative magnitudes in absence of external forces. Independently of the interaction among the bodies of the same set these magnitudes evaluated over all members of the group do not change.

Gyration radius

One of the parameters most widely used to characterise an agglomerate is the gyration radius $R_g(\vec{r})$ around a generic axis $\vec{r}$. The square of the gyration

[2]The Kronecker's delta δ_{ij} takes value 1 when $i = j$ and 0 when $i \neq j$.

radius around an axis $\vec{r}$ (which contains the centre of mass) is defined as the quotient between the inertia around this axis over the total mass

$$R_g^2(\vec{r}) = I(\vec{r})/m. \tag{4.48}$$

This is the gyration radius calculated from 2D photographies obtained by microscopy, when a projection is used in the absence of the 3D structure. In this case the vector $\vec{r}$ follows the line of sight of the camera. With the 3D structure available a generic inertia matrix can be calculated. By diagonalisation the inertia matrix can be decomposed in the product

$$[I] = [R][\Lambda][R]^{-1} \tag{4.49}$$

where $[\Lambda]$ is a diagonal matrix which contains the eigenvalues of the inertia matrix and $[R]$ the rotation matrix whose columns are the corresponding eigenvectors. Each of the eigenvalues is one of the principal components of the inertia matrix λ_{ii} and it is associated with the eigenvector $\vec{r}_i$ which follows the i-th principal direction. The eigenvalues are sorted in increasing order. A gyration radius exists associated to each principal direction

$$R_{g,i}^2 = \lambda_{ii}/m. \tag{4.50}$$

The difference between the main gyration radii is related with the asymmetry of the agglomerate. When $\lambda_{11} \simeq \lambda_{22} \simeq \lambda_{33}$ the agglomerate is spherical. If $\lambda_{11} + \lambda_{22} \simeq \lambda_{33}$ the agglomerate is planar. Finally, if $\lambda_{11} \ll \lambda_{22} \simeq \lambda_{33}$ the agglomerate is lineal. The inertia moment respect to the centre of mass is related with the inertia moment respect to the main axes by the expression

$$2I_0 = \lambda_{11} + \lambda_{22} + \lambda_{33}. \tag{4.51}$$

A gyration radius associated to the moment of inertia respect to the centre of mass can be calculated from the previous values as

$$R_{g,0}^2 = I_0/m = (\lambda_{11} + \lambda_{22} + \lambda_{33})/2m. \tag{4.52}$$

This value is related with the expected value calculated from the 2D projection of an agglomerate (obtained from 2D photographies). The expected value for the inertia projected on a generic axis is $\mathbb{E}[I(\vec{r})] = (\lambda_{11} + \lambda_{22} + \lambda_{33})/3$. Then the observed 2D expected gyration radius is related with $R_{g,0}$ by the expression

$$3\mathbb{E}[R_g^2(\vec{r})] = 2R_{g,0}^2 \tag{4.53}$$

Considering spherical particles the inertia matrix using Einstein convention is

$$I_{ij} = \frac{4}{3}\pi\rho R_n^3 \left[\left(\frac{2}{5}R_n^2 + x_{n,k}^2 \right) \delta_{ij} - x_{n,i}x_{n,j} \right] \tag{4.54}$$

where R_n is the radius of the n-th precursor, $\vec{x}_n$ is its relative location with respect to the centre of mass, ρ is the density of the material[3] and δ_{ij} is the Kronecker's delta. Every inertia moment has been calculated following the expression 4.54. When $R_n \ll R_{g,k}$ the inertial contribution of the spherical shape of each precursor is negligible and the expression can be approximated by

$$I_{ij} = \frac{4}{3}\pi\rho R_n^3 \left(x_{n,k}^2\delta_{ij} - x_{n,i}x_{n,j}\right). \tag{4.55}$$

This approximation is usually employed to calculate the gyration radii by other authors. The total mass is calculated as the sum of the masses of each precursor

$$m = \frac{4}{3}\pi\rho R_n^3. \tag{4.56}$$

Fractal dimension

Agglomerates present self-similarity between different scales. The dependency of the agglomerate size (gyration radius $R_{g,0}$) with the number of precursors n_p follows an exponential law

$$n_p = k_f \left(R_{g,0}/R_p\right)^{D_f}. \tag{4.57}$$

Here k_f is a premultiplicative constant, R_p is the precursor characteristic radius and D_f is an exponent called fractal dimension of the agglomerate. It is an indirect measurement of the compactness of the agglomerate. There is a self-similarity relation between scales. The proportion of filled regions is constant independently of the level of detail.

The value of the fractal dimension of the resultant population can depend on the flow regime and the generation algorithm. Meakin [12] studied multiple algorithms and analysed the results. He shows how the interaction between the aggregates can determine its shape. Similarly Dekkers and Friedlander [11] study the fractal dimension of computer generated particles. They created them by Brownian motion. No interaction force has been considered in this thesis for the generation process.

[3]The density of the material is not needed to calculate the gyration radius. In such cases it is recommended to replace ρ by $3\pi/4$ to reduce the number of required operations.

Agglomeration

The agglomeration is the process which fuses two or more agglomerates into a single one. Solving this problem is as easy as solving the equations

$$\sum_{n=1}^{N} m_n \vec{u}_n = m^f \vec{u}^f \qquad \text{and} \qquad \sum_{n=1}^{N} [I_n^0]\vec{\omega}_n + m_n \vec{d_n} \wedge \vec{u}_n = [I^{0^f}]\vec{\omega}^f \tag{4.58}$$

where the relative location $\vec{d_n}$ is the distance from the new centre of gravity $\vec{x}^f$ to each of the old ones $\vec{x}_n$.

Collision

The collision process is usually calculated from the coefficient of restitution C_R. It relates the relative velocity of one particle respect to other $\vec{u}_{ij} = \vec{u}_j - \vec{u}_i$ projected into the normal $\vec{n}$ to the surface of the i-th body before $\vec{u}_{ij}$ and after $\vec{u}^*_{ij}$ the collision

$$\vec{u}^*_{ij} \cdot \vec{n} = -C_R \, (\vec{u}_{ij} \cdot \vec{n}). \tag{4.59}$$

In an indirect way, the coefficient of restitution is a characteristic of the material which measures the amount of energy remaining after an impact between two rigid bodies. When the coefficient of restitution is 1, the energy after the impact is conserved. On the contrary, if the coefficient of restitution is 0, the energy dissipation is maximum. Usually the collision process is solved by the calculation of the momentum exchange. However, due to robustness reasons, a methodology based on energy has been implemented. The kinetic energy of a body is the sum of its kinetic energy due to translation plus the kinetic energy due to rotation

$$E = \frac{1}{2} m \vec{u}^T \vec{u} + \frac{1}{2} \vec{\omega}^T [I] \vec{\omega}. \tag{4.60}$$

Accordingly to the second thermodynamic law after the collision of two agglomerates the remaining energy has to be lower or equal than at the beginning $\sum E_n \geq \sum E^*_n$. If the collision solver is based on momentum exchange the inequation could not be accomplished due to numerical errors. On the contrary, if the total energy after collision is imposed and then the momentum exchange is calculated, the over-energising effects are completely avoided. How much energy is lost in the process is fixed by the square of the restitution coefficient. When this value is 0 the remaining energy, a minimum, is the energy contained in the fused agglomerate. The agglomerate is obtained solving the equation system proposed in equation 4.58. On the contrary, if the value is 1, the energy is conserved. The intermediate values are obtained from expression

$$E^* = C_R^2 \left(E - E^f \right) + E^f \tag{4.61}$$

where no superscript means before collision, the superscript ^{f} means after the agglomeration process and * means after collision. If $\vec{x}^i$ is the impact point and $\vec{d}_1^i$ and $\vec{d}_2^i$ the relative location of the same point respect to the centre of gravity of each colliding agglomerate then it can be calculated the velocity of this point for both particles

$$\vec{u}_1^i = \vec{u}_1 + \vec{\omega}_1 \wedge \vec{d}_1^i \qquad \text{and} \qquad \vec{u}_2^i = \vec{u}_2 + \vec{\omega}_2 \wedge \vec{d}_2^i. \tag{4.62}$$

The next step is to calculate the normal. Because the agglomerates are formed by spheres the normal is something very easy to be calculated. Once it is known which spheres are in contact the normal is computed as the distance between the centres of the colliding precursors divided by its own modulus

$$\vec{n} = \left(\vec{x}_2^p - \vec{x}_1^p\right) / \left\|\vec{x}_2^p - \vec{x}_1^p\right\|. \tag{4.63}$$

It has to be noticed how the particles only collides if the intersecting precursors are getting closer. Mathematically the condition for collision is

$$\left(\vec{u}_2^i - \vec{u}_1^i\right) \cdot \vec{n} < 0. \tag{4.64}$$

The distance to the precursors can be also used to calculate the impact point. If the collision is compute at the end of each time step and not at the middle of it there is an intersection region instead of a single contact point. A fast approximation can be easily calculated as

$$\vec{x}^i = \left(\vec{x}_2^p r_1^p + \vec{x}_1^p r_2^p\right) / \left(r_1^p + r_2^p\right) \tag{4.65}$$

where r_1^p and r_2^p are the radii of the colliding precursors. The collision produces an impulse ΔP in the point $\vec{x}^i$ coming from a force whose direction follows the normal (no friction), so the vector form is $\Delta P\vec{n}$ on the first agglomerate and the negative counterpart in the second one. Because the impulse is not necessary aligned with the centre of gravity of each agglomerate it also produces variations in the angular momentum $\Delta\vec{L}_1 = \vec{d}_1^i \wedge \Delta P\vec{n}$ and $\Delta\vec{L}_2 = -\vec{d}_2^i \wedge \Delta P\vec{n}$. The linear and angular velocities after collision can be calculated from the previous values as

$$\vec{v}_1^* - \vec{v}_1 = m_1^{-1}\Delta P\vec{n} \qquad \text{and} \qquad \vec{\omega}_1^* - \vec{\omega}_1 = [I_1]^{-1}\left(\vec{d}_1^i \wedge \Delta P\vec{n}\right). \tag{4.66}$$

The total post-collision energy can be expressed as a second order function of the impulse magnitude ΔP.

$$2E^* = a_0 + 2a_1\Delta P + a_2\Delta P^2 \tag{4.67}$$

The parameters are

$$a_0 = m_1\vec{v}_1 \cdot \vec{v}_1 + m_2\vec{v}_2 \cdot \vec{v}_2 + \vec{\omega}_1[I_1]\vec{\omega}_1 + \vec{\omega}_2[I_2]\vec{\omega}_2, \tag{4.68a}$$

$$a_1 = \vec{v}_1 \cdot \vec{n} - \vec{v}_2 \cdot \vec{n} + \vec{\omega}_1 \cdot (\vec{d}_1 \wedge \vec{n}) - \vec{\omega}_2 \cdot (\vec{d}_2 \wedge \vec{n}), \tag{4.68b}$$

$$a_2 = m_1^{-1} + m_2^{-1} + (\vec{d}_1 \wedge \vec{n})[I_1]^{-1}(\vec{d}_1 \wedge \vec{n}) + (\vec{d}_2 \wedge \vec{n})[I_2]^{-1}(\vec{d}_2 \wedge \vec{n}). \tag{4.68c}$$

The solution of the second order equation is

$$\Delta P = \frac{-a_1 \pm \sqrt{a_1^2 - (a_0 - 2E^*)a_2}}{a_2}. \tag{4.69}$$

Once the impulse is known the linear and angular velocities of the agglomerates after the collision are calculated by the equation 4.66.

Collision vs. agglomeration probability

When two agglomerates are in contact they can collide and change their motion or they can fuse to create a larger agglomerate. A criterion decides the way to be followed. For the studied cases this criterion is based on the energy absorbed during the collision. If the agglomerates fuse the link formed at the contact point has to absorb all the energy excess. Consequently it is expected that the probability to fuse the impinging agglomerates is higher when the energy to be absorbed by the new link is low. On the contrary, if the energy to be absorbed during the fusion process is too high is much more probable the collision than the agglomeration. Coherently to the previous observations a probability function is proposed to decided when an agglomerate collides and when it fuses

$$\mathscr{P}(\Delta E) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\log_{10}(\Delta E) - \log_{10}(\Delta E_{1/2})}{\sqrt{2}\log_{10}(\Delta E_\sigma)}\right)\right]. \tag{4.70}$$

The logarithmic scale is mandatory to force agglomeration when the energy to be absorbed during the impact is zero. Here $\Delta E_{1/2}$ is the energy when the probabilities to collide and fuse are the same and ΔE_σ is the parameter which controls how abrupt is the transition. When ΔE_σ is small the zone around $\Delta E_{1/2}$ where collisions and agglomerations coexist is also small. In the limit, when $\Delta E_\sigma \simeq 0$, the probability function $\mathscr{P}(\Delta E)$ is the unit step function. All the impacts whose absorbed energy for agglomeration is lower than $\Delta E_{1/2}$ results in fusion whereas greater values ends in collision.

Once the probability function is known a random sample from a uniform distribution χ decides when the impinging agglomerates fuse and when they collides. The criterion to collide is evaluated every time that two agglomerates are in contact by the condition $\mathscr{P}(\Delta E) > \chi$.

4.3.2 Algorithms: impact detection

An agglomerate is formed by spherical precursors. The number of these precursors can differ a lot from one agglomerate to other. For the studied cases,

this number goes from a single particle in the simplest case to tens of thousands for the largest particles or even more. In other words, there is a discrepancy in scale, complexity and consumption of computational resources. Multi-scale interactions are the curse of computational parallelisation due to the bad load balance among processors. A little portion of the total number of interactions consumes most of the resources. The number of operations required to detect when two agglomerates are close to each other is of order $\mathscr{O}(N^2)$, where N is in this context the number of agglomerates in the domain.

When proximity is detected it is time to check if the collision exists. During this process the distance between precursors of different agglomerates is calculated one by one. The brute force algorithm required to detect impact is proportional to the product of the number of precursors of each colliding agglomerate $\mathscr{O}(N_1 N_2) \simeq \mathscr{O}(N^2)$. Here N_1 and N_2 are the number of precursors of each agglomerate and its product is the number of binary combinations (a precursor from the first agglomerate can collide with every precursor of the second one). Consequently large variations of concentration and size of agglomerates are penalties to load balance. In this work several algorithms are applied for the same purpose depending on these variables for optimising. The computer used for calculations is a workstation, a shared memory machine. OpenMP is the library used for parallelisation. The algorithms have been adapted in consequence.

Two of these algorithms are the proximity detector and the contact detector. Both of them have as purpose to decrease the order of their original brute force counterparts. The reduction goes from order $\mathscr{O}(N^2)$ to $\mathscr{O}(N \log(N))$. Divide and conquer algorithms are the responsible of this improvement. They are detailed in the following lines.

Proximity detector

Testing when the collision happens is a very expensive task, even if an advance and optimum algorithm is used instead of the brute force one. The complexity of this process grows with the size of the agglomerates to be tested. It becomes prohibitive if this number is very large or if they are formed by a great number of precursors. The fastest way to prevent such huge demands of resources is just to skip the search of impacts by means of a rejecting criterion. For this purpose each agglomerate has been surrounded by an sphere centred in the centre of mass of the own agglomerate. The collision could happen if and only if the surrounding spheres to the colliding agglomerates intersect each other. The radius R of each sphere has to be big enough to contain whole the volume of the agglomerate

$$R = \max\left(\|\vec{x}_i - \vec{x}_{cm}\| + r_i\right), \tag{4.71}$$

where $\vec{x}_i$ is the location of the centre of the i-th precursor, r_i its radius and $\vec{x}_{cm}$ is the centre of mass of the agglomerate. Larger improvements could be achieved if smaller circumscribed spheres are used instead. However important differences between both spheres can only be observed in large sparse agglomerates, uncommon in this context. The penalty to be paid for the calculation of the optimum sphere counteracts the advantages of its use. Consequently and for the sake of simplicity the algorithm to find the optimum sphere has not been implemented and the circumscribed sphere centred in the centre of mass has been used instead.

Two agglomerates are considered candidates to collide when the surrounding spheres intersect each other. It happens when the distance between the centres of mass is lower than the sum of their radii[4]

$$(R_1 + R_2)^2 \geq \|\vec{x}_{cm,1} - \vec{x}_{cm,2}\|^2 . \tag{4.72}$$

Using this criterion the computational time is drastically reduced. However the procedure is still very expensive, specially when the number of agglomerates is very large. Each agglomerate can collide with every other. The number of times the criterion has to be evaluated grows proportionally to the square of the number of agglomerates. However full comparison is not necessary. A divide and conquer strategy reduces again the complexity of the algorithm to order $\mathscr{O}(N \log N)$, where N is the number of agglomerates in the domain.

The divide and conquer algorithm starts from the division of the original domain in several regions. At this moment the regions are equal (cubes) and structured but more advanced algorithms can be used if wanted. Each of this regions is surrounded by a sphere whose radius is the distance from the centre of the sub-domain to the furthest corner. If the sub-domain is cubic the radius of its surrounding sphere is $R^{(0)}_{D\&C} = \sqrt{3}h^{(0)}_{D\&C}/2$, where $h^{(0)}_{D\&C}$ is the length of the sub-domain. Once the domain is divided it is time to select which agglomerates intersect them. The condition to include an agglomerate in a sub-domain is similar to the criterion to select when two agglomerates could collide (see equation 4.72). An agglomerate is included in the sub-domain if the surrounding spheres intersect each other. The condition is satisfied when the sum of the radii is less than the distance between the centres

$$\left(R_{ag} + R^{(0)}_{D\&C}\right)^2 \geq \|\vec{x}_{D\&C} - \vec{x}_{cm}\|^2 . \tag{4.73}$$

Each of these sub-domains can be calculated by a different core. The cost of this operation is MN, where M is the number of sub-domains and N the total number of precursors. Resulting of the decomposition there are M parts with $N^{(0)} \sim N/M$ agglomerates each. The computational cost required to

[4]The reason to use the square instead of directly the magnitude is to avoid the computationally expensive calculation of the square root.

solve one of these parts is ${N^{(0)}}^2$. Once the domain has been divided in regular blocks, each block can also be divided recursively into self-similar but smaller sub-blocks following a geometrical progression with each step. The common ratio k which relates two levels of refinement $h^{(1)}_{D\&C} = kh^{(0)}_{D\&C}$ is the inverse of a positive integer number called refinement factor $K = 1/k$. With this strategy each block is divided in K^3 sub-blocks. The computational cost of solving each block tens to zero with its size, and the computational cost to divide the l-th level is the number of blocks MK^{3l} of each level times the computational cost of dividing a single block $NM^{-1}K^{-3l+3}$. The total cost per level is linear and it does not depend of the level itself NK^3. Because this operation has to be repeated for each level and the number of required levels L scales with the logarithm of the number of agglomerates $L \propto \log_{K^3}(N)$ the number of operations is proportional to $MN + K^3 N \log_{K^3}(N)$. It corresponds with order $\mathcal{O}(N \log(N))$.

All the behaviour described before works under the assumption of punctual agglomerates. However the incorrectness of this hypothesis has negative consequences on the performance. Volumetric agglomerates can occupy several blocks. During the simulation some of the resulting agglomerates extend over large parts of the domain. As a result, if two of this particles interacts, that interaction could be evaluated multiple times resulting in a penalty of consumption of computational resources.

Impact detection

Once the possibility of collision has been confirmed the algorithm for impact detection takes place. As usual, if the sum of the radii of the precursors is greater than the distance between the centres the collision exists

$$(r_{i_1} + r_{i_2})^2 \geq \|\vec{x}_{i_2} - \vec{x}_{i_1}\|^2 . \tag{4.74}$$

Each i_1-th precursor of the first agglomerate could collide with every i_2-th precursor of the second. Consequently $N_1 N_2$ operations are necessary. The resources demand grows with the square of the agglomerates size. One way to reduce the number of required operations is to take advantage of the spheres which delimit the area of influence around an agglomerate. Every precursor of the second agglomerate candidate to collide with the first agglomerate has to be contained, at least partially, in the sphere which surrounds the first particle. This condition can be used as filter, a precursor of the second agglomerate can collide with the first particle if and only if

$$(R_1 + r_{i_2})^2 \geq \|\vec{x}_{cm,1} - \vec{x}_{i_2}\|^2 , \tag{4.75}$$

otherwise any combination of the i_2-th precursor of the second agglomerate with every precursor of the first agglomerate is not a candidate to collide and

it can be rejected. The same procedure can be done in the opposite way. However using both conditions at the same time is only functional for interactions between large agglomerates. For small-large or small-small interactions it is better to delimit the volume with the smallest sphere (usually the volume of influence of the smallest particle is contained in the volume of influence of the largest one).

The number of required operations can be still great after rejecting the furthest precursors. Similarly to previous section a divide and conquer algorithm has been implemented to accelerate the calculus. Taking the surrounding sphere as reference the volume is divided recursively reducing the population contained in each block. When the number of required operations is under a threshold (product between the number of precursors of the first and second agglomerate contained in the block) every possible combination is calculated with expression 4.74.

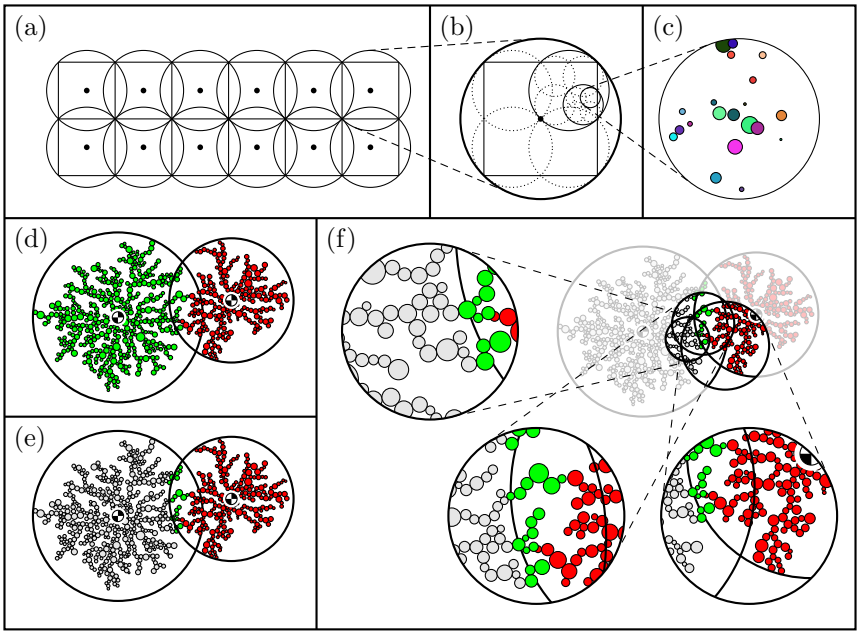

Figure 4.5: Collision detection algorithm. (a) Division of the domain in blocks. (b) Recursive division of each block in sub-blocks. (c) Looking for impact candidates. (d) Detection of intersection between areas of influence. (e) Discarding precursors of the largest agglomerate if they are outside of the area of influence of the smallest agglomerate. (f) Recursive division of the area of influence for the smallest agglomerate and impact detection.

4.3.3 Non-dimensional descriptors for populations

Several hypotheses with respect to precursor populations reduce the complexity of the simulation. The precursors are supposed to be spherical, they follow a log-normal distribution in size and Maxwellian in velocity, both of them uncorrelated, their concentration is considered uniform in space and the discrimination between collision and agglomeration after contact is controlled by an error function dependent on the energy absorption required for agglomeration. Then the collision between precursors is controlled by seven parameters:

- concentration of precursors n,
- velocity deviation[5] c_σ,
- characteristic precursor radius r_μ (the median of the log-normal distribution),
- deviation of the logarithm of the precursor radius $\log(r_\sigma)$,
- agglomeration energy absorption for equal probability between collision and agglomeration $\Delta E_{1/2}$,
- deviation of the logarithm of the agglomeration energy absorption of the probability function $\log(\Delta E_\sigma)$,
- the restitution coefficient for impacts C_R (see section 4.3.1 for more details).

Dimensionless parameters replace the previous magnitudes. The adimensionalisation is carried out taking as reference the characteristic precursor radius r_μ, the deviation of the speed c_σ and the density of the material ρ. Every other magnitude can by scaled by a combination of them. From now in advance every dimensionless magnitude will be denoted by the superscript *. A generic variable q is related with its dimensionless counterpart q^* by the scale factor S as $q = Sq^*$. For example, the dimensionless radius is $r^* = r/r_\mu$, the dimensionless velocity $c^* = c/c_\sigma$ and the dimensionless time $t^* = tc_\sigma/r_\mu$. The rest of the magnitudes can be adimensionalised in analogous way.

Two advantages come with adimensionalisation. Firstly it is gained in generality. It can be taken advantage of self-similarities to join in a single dimensionless result multiple combinations of physical analogous values. The results become independent to any physical scale. The other advantage is the reduction of the number of variables. Some of the physical magnitudes are

[5]The collision formulation is based on the relative speed so it is insensible to the mean velocity of the Maxwellian distribution.

joined into a single non-dimensional parameter. As a result the number of required experiments to complete the sensibility study is reduced. For the case of interest, two of the seven variables becomes constant in dimensionless space, the characteristic precursor radius $r_\mu^* = 1$ and the velocity deviation for each direction $c_\sigma^* = 1$. Because the complexity of the sensibility analysis grows exponentially with the number of variables this little reduction saves an important amount of computational effort. This is what is called the course of dimensionality.

Non-dimensional descriptors sampling: Latin-Hypercube

The described agglomeration process is almost stochastic. Most of the parameters which control the process, from the precursors generation to the discrimination between collision and agglomeration, are generated by random sampling accordingly to PDF. Consequently dispersion is expected in the solution. The same combination of input parameters returns a population of outputs instead of a single value. This is an inconvenient for sensibility analysis. The discrepancies between two samples could be explained by the differences in the inputs but also by the natural discrepancy due to chaos (populations instead of single values). Under these conditions the number of required simulations has to satisfy two criteria. Firstly, it has to be large enough and variate enough to cover the effects induced by every input parameter. Secondly, the number of samples has to be large enough to obtain a characteristic population or, in other words, the number of samples has to be large enough to smooth the effects of the uncertainty. If it is not possible the use of spectral methods to characterise the uncertainty, like this case, sampling is the alternative. Multiple options exist. Latin-Hypercube has been considered the best option for the studied case. This method covers the whole range of variability of the input variables and reduces the occurrence in comparison to other techniques like Monte-Carlo sampling, avoiding clustering of samples and empty spaces.

Several variations of the Latin-Hypercube algorithm exist. In this article the simplest case has been implemented. The samples of every input parameter q are bounded by an interval (q_{min}, q_{max}). Once the interval has been established it is time to decide how many samples M are required. The interval is divided in M equal-probable subintervals of length $\Delta q = (q_{max} - q_{min})/M$. Then the m-th subinterval is $(q_{min} + (m-1)\Delta q, q_{min} + m\Delta q)$. A random sample is extracted from a uniform distribution for each interval and variable. At this time there are M input values to run M simulations, sorted in increasing order. To obtain correct results the input parameters have to be uncorrelated. It does not happen if they are sorted. The next step is breaking the order with random permutations. Once each combination is uncorrelated it can be used as input for each simulation.

4.3.4 Simulations

Space and time dependence are studied. Two sets of simulations have been done. For time dependence simulation, a periodic domain is studied keeping the same precursors from the beginning. How the agglomerate population changes in space is studied by the simulation of a region connected to a reservoir of precursors.

Variations of agglomerate populations in time

Starting for an initial population of free precursors into a full-periodic domain this simulation shows how the population of agglomerates changes in time. The population of precursors follows a log-normal distribution in size and the velocity is characterised by a Maxwell's distribution. The simulation finishes when the number of agglomerates is lower than 128. One of the results can be seen in figure 4.6.

The simulation set covers 128 cases, each one generated randomly using the Lattice-Hypercube technique. The random samples are uniform distributed and they cover 5 variables. The table 4.1 shows the variables and their limits.

Variable	**Mathematical expression**	**Lower Limit**	**Upper Limit**
Concentration	$\log_{10}(\quad n \quad)$	-6.0	-5.0
Restitution coefficient	$\log_{10}(1 - C_R^2)$	-3.0	0.0
Variation of particle size	$\log_{10}(\quad \sigma_S \quad)$	0.1	1.0
Collision/Agglomeration discriminant	$\log_{10}(\Delta E_{1/2})$	0.0	4.0
Collision/Agglomeration coexistence area	$\log_{10}(\quad \Delta E_\sigma \quad)$	0.1	1.5

Table 4.1: Parameters and their variations in the study.

After simulating every case the final populations have been compiled and represented. Figure 4.7 shows the relation gyration radius *vs.* number of precursors per agglomerate. It can be seen how there is no dependence of the agglomerate structure with respect to any of the variables of the study. Every agglomerate is located in the surroundings of the same tendency line independently of the input parameters of the simulation. The fractal dimension which relates the number of precursors with the gyration radius of the agglomerates has been obtained in a similar way to other researchers [5]

$$n_p = k_p \left(R_g/R_0\right)^{D_f} \tag{4.76}$$

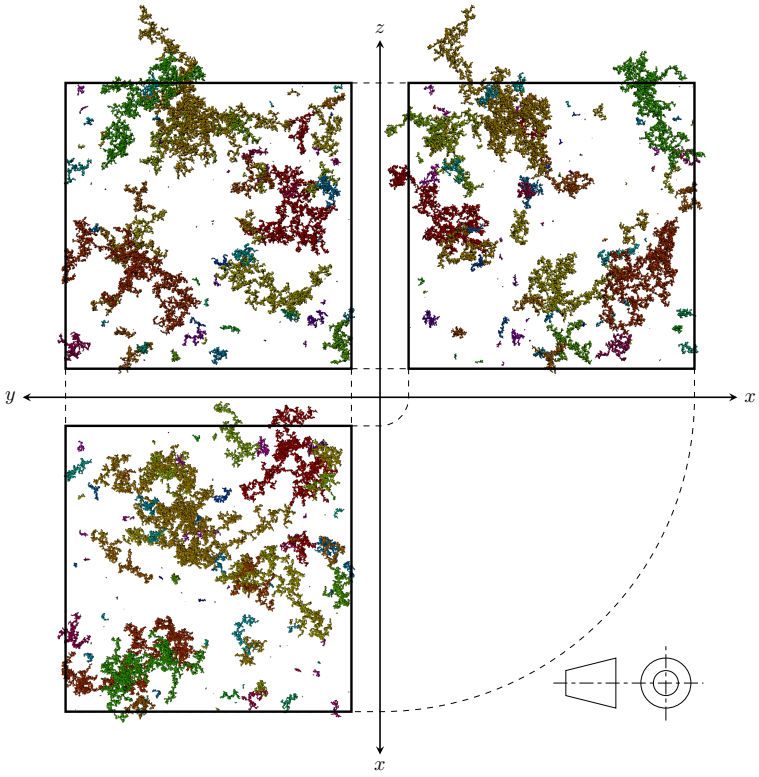

Figure 4.6: Population of agglomerates at the end of a simulation.

except two little differences. Usually it is considered the agglomerate much larger than the precursor. Under this hypothesis the precursors can be considered as point masses. This simplification is valid only for large agglomerates, when the size of the agglomerate is much larger than the size of a single precursor. However this is not truth when the agglomerate is made up of few precursors. To avoid the distortion to unrealistic results for small agglomerates the 3D effects of every single precursor has been considered. The second modification is introduced for the approximation of the tendency line. Other authors calculate the tendency line applying directly mean least squares. The samples presented here are not equally distributed. Consequently some regions contributes differently depending on the local concentration of samples. A weighting function is proposed to compensate these effects. Every tram of samples with respect to the number of precursors has to contribute equally for the tendency calculation. To achieve this objective the proposed weight-

ing function has to be the inverse of the concentration. The first step is then to obtain the concentration as a function of the logarithm of the number of precursors. It is propose to use a set of points where the values of the concentration will be known, called nodes from now in advance. The values between nodes are calculated by interpolation. Every n-th sample contributes to the concentration function by the kernel

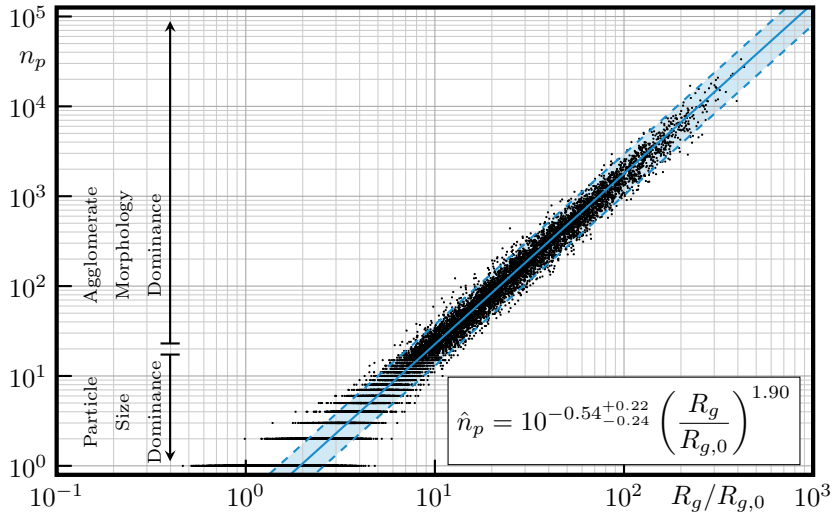

Figure 4.7: Relation between the gyration radius and the number of precursors. Population of samples (black spots) and tendency line (blue continuous line) with thresholds (blue dashed line) for 95% of confidence interval (blue area).

$$K(\xi, \xi_n) = \exp\left[-(\xi - \xi_n)^2/2\sigma^2\right] \tag{4.77}$$

where $\xi = \log(n_p)$ in this context. The addition of every contribution returns the concentration function

$$\rho(\xi) = \sum_n \exp\left[-(\xi - \xi_n)^2/2\sigma^2\right] \tag{4.78}$$

The values of the i-th node is then

$$\rho_i = \rho(\xi_i) = \sum_n \exp\left[-(\xi_i - \xi_n)^2/2\sigma^2\right] \tag{4.79}$$

The presence of a sample affects the surroundings of ξ_n depending on the value of σ. The main objective of σ is to smooth this noise getting a smooth and continuous density function $\rho(\xi)$. The larger the value of σ, the smoother solution. Once the density function is calculated it is time to compute the

weighting function. The weighting function is defined as the inverse of the density function for every node $w_i = 1/\rho_i$. The intermediate values $w(\xi)$ can be interpolated by Tayloring the surrounding nodes and solving the resultant equation system. Being $\Delta\xi_i = \xi_i - \xi$, the value of w_i can be approximated as

$$w_i = \sum_{n=0}^{\infty} \frac{{\Delta\xi_i}^n}{n!} \partial_\xi^{(n)} w(\xi). \tag{4.80}$$

The series is truncated at the desired order N

$$w_i = \sum_{n=0}^{N} \frac{{\Delta\xi_i}^n}{n!} \partial_\xi^{(n)} w(\xi) + \mathscr{O}({\Delta\xi_i}^N). \tag{4.81}$$

Employing $N+1$ nodes where the weighting function is known and solving the resultant equation system the approximation of $w(\xi)$ and its first N derivatives are obtained

$$\begin{bmatrix} 1 & S\Delta\xi_0 & (S\Delta\xi_0)^2 & \cdots & (S\Delta\xi_0)^N \\ 1 & S\Delta\xi_1 & (S\Delta\xi_1)^2 & \cdots & (S\Delta\xi_1)^N \\ 1 & S\Delta\xi_2 & (S\Delta\xi_2)^2 & \cdots & (S\Delta\xi_2)^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & S\Delta\xi_N & (S\Delta\xi_N)^2 & \cdots & (S\Delta\xi_N)^N \end{bmatrix} \begin{Bmatrix} w(\xi) \\ \partial_\xi w(\xi)/S \\ \partial_\xi^2 w(\xi)/2S^2 \\ \vdots \\ \partial_\xi^N w(\xi)/N!S^N \end{Bmatrix} = \begin{Bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{Bmatrix}. \tag{4.82}$$

The multiplicative factor $1/N!S^N$ is included in the solution. It can be observed how the values to be solved are $\partial_\xi^N w(\xi)/N!S^N$ instead of obtaining directly the derivative. In first place, including $N!$ as part of the unknown reduces the number of required operations. Secondly, the inclusion of both $N!$ and S as part of the unknown improves the condition number of the matrix, reducing the numerical error and the number of iterations needed to converge. S is a scale factor which satisfies $1 \approx S\Delta\xi_i$. The interpolation is the first component of the solution. Figure 4.8 shows the density function and its inverse, the weighting function, in random units. It can be observed how their values are symmetric in logarithmic scale (the mirror line is not located at 0 because the functions have been normalised after the calculus).

Once the weighting function has been established the weight for each agglomerate can be calculated as a function of the number of precursors. The obtained results are the weights to be used to calculate the tendency line. The employed method is the weighted mean least squares. Applying the weighted mean least squares method, the results of the previous step are the weights to be used to compute the tendency line. However there is still a problem. After plotting the samples (see figure 4.7) two regions can be distinguish. When the agglomerates are formed by a low number of precursors the gyration radius is dominated by the 3D effects of the precursors (precursor size). On the contrary, when the hypothesis of point precursors becomes realistic, the tendency

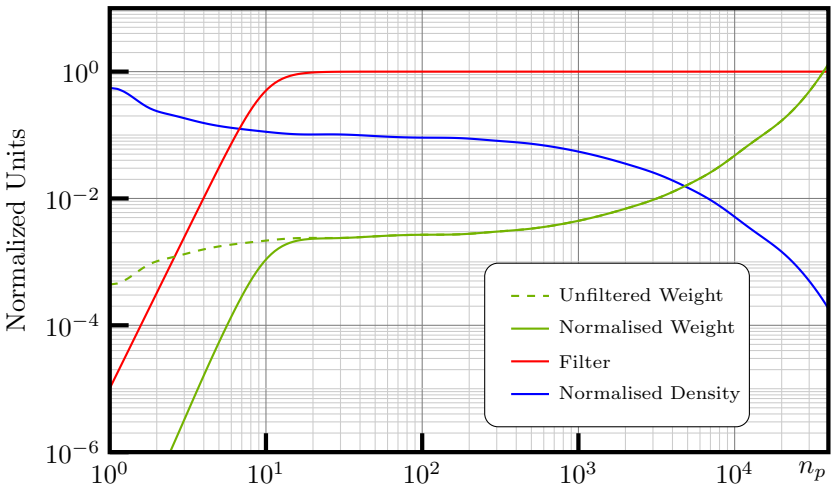

Figure 4.8: Density function (blue), filter (red), weight (dashed green) and filtered weight (solid green) in random units as function of number of precursors.

is dominated by the agglomerate morphology. We are interested on the study of the morphology, so the first part of the population has to be filtered to avoid the disturbance of the results. A logistic curve is the responsible of this task

$$f(\xi) = \{1 + \exp[-5(\xi - 2.3)]\}^{-1}. \tag{4.83}$$

The filtered weight is obtained by the multiplication of the filter by the inverse of the density of samples $\hat{w}(\xi) = f(\xi)/\rho(\xi)$. Applying the weighted mean least squares method with this new weight the fractal dimension is corrected to take more in consideration large particles.

But the fitting line does not approximate perfectly the population of particles. There is uncertainty. Several agglomerates with different number of particles can return the same gyration radius. To describe properly the entire population the uncertainty has to be quantified. This is done by the study of the distribution of the residuals. The residual of each sample is calculated as the difference between the expected value given by the tendency line and the real value of the sample

$$\varepsilon_n = \log(n_{p_n}) - \log[\hat{n}_p(R_{g_n})]. \tag{4.84}$$

In similar way to the job done to create the density function, a kernel function 4.77 is applied to calculate the contribution of each sample around each residual. With this strategy the error distribution function is smoothed. In addition, to obtain congruent results with previous work, the error function

has to be weighted equally than mean least square method

$$\mathrm{pdf}(\varepsilon) = \frac{1}{K}\sum_{n=1}^{N} \hat{w}(\log(n_{p_n}))\exp\left[-\frac{(\varepsilon-\varepsilon_n)^2}{2\sigma^2}\right]. \tag{4.85}$$

The total area can be obtained by the integration of the previous equation. The function is a PDF so the value under the curve is unitary. The premultiplicative factor K can be calculated

$$K = \int_{-\infty}^{\infty}\sum_{n=1}^{N} \hat{w}(\log(n_{p_n}))\exp\left[-\frac{(\varepsilon-\varepsilon_n)^2}{2\sigma^2}\right] d\varepsilon. \tag{4.86}$$

Both the PDF and the CDF are represented in figure 4.9.

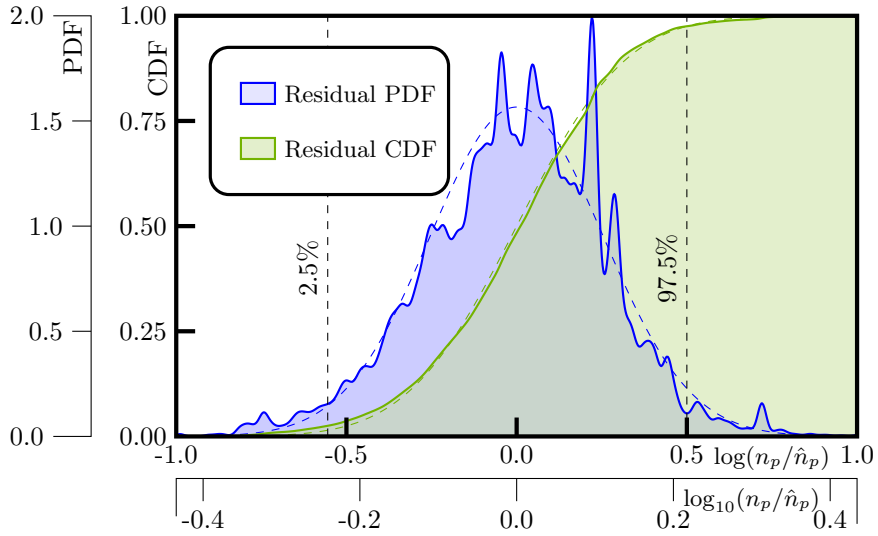

Figure 4.9: Probabilistic Density Function (blue) and Cumulative Distribution Function (green) of residuals with respect to the logarithm of the number of precursors per agglomerate (continuous shaded line) and its best fitting normal distribution (dashed line)

Evolution of the number of agglomerates in time

The evolution of the number of agglomerates with time has been approximated starting by a modified exponential decay

$$n_p = n_{p,0}\exp\left[-\left(t/\tau_0\right)^{\beta(t)}\right]. \tag{4.87}$$

The exponent $\beta(t)$ has been included to take into account the effects of the agglomerate growth. After plotting the exponent, calculated by the expression

$$\beta = \log\left[-\log\left(n_p\right)\right] / \log\left(t/\tau_0\right), \tag{4.88}$$

some properties have been observed (see figure 4.10):

- At the beginning, when there are only precursors (agglomerates with a single precursor), the decaying quantity follows the conventional law. In other words, $\beta(0) = 1$ (See detail of figure 4.10).
- The value of $\beta(t)$ decays asymptotically to a fixed value $\lim_{t\to\infty} \beta(t) = c_0$. The asymptotic behaviour has been established including an exponential in the approximation $\hat{\beta}(t) = c_0 + (1 - c_0)\exp(-(t/\tau_1)^{\alpha_1})$. This expression satisfies both the present and the previous conditions.
- At intermediate times there are a lot of discrepancies between the cases. Sometimes the decaying velocity increases until a local maximum to decrease afterwards. Other times β decreases monotonically. A power law is a good candidate to emulate every behaviour. An approximation of β which satisfies every condition is

$$\hat{\beta}(t) = c_0 + \left[(1 - c_0) + c_1\left(\frac{t}{\tau_1}\right)^{\alpha_0}\right]\exp\left[-\left(\frac{t}{\tau_1}\right)^{\alpha_1}\right]. \tag{4.89}$$

The evolution of the number of agglomerates in time can be approximated by the equation 4.87. Each simulation returns a different set of optimum-fitting constants. The followed criteria to fit the function is the minimisation of the mean quadratic error of exponent β with respect to the dimensionless time $t^* = t/\tau_0$,

$$\phi = \frac{1}{t^*_{end}} \int_0^{t^*_{end}} \left[\beta\left(t^*\right) - \hat{\beta}\left(t^*\right)\right]^2 dt^*, \tag{4.90}$$

where t^*_{end} is the dimensionless simulated time. The figure 4.11 resumes the results. The error in the number of precursors is bounded by the interval -0.015 and 0.003 times the initial number of precursors. The next objective is to find the relation between the fitting parameters and the variables which control the physics.

It is supposed that every fitting factor is linear-dependent on the random-generated physical parameters. The approximation for every case is stored in a matrix $[Y]$, where $[Y]_{n,i}$ contains the i-th fitting factor of the n-th simulation. Similarly the values of the physical parameters are stored in the matrix $[U]$.

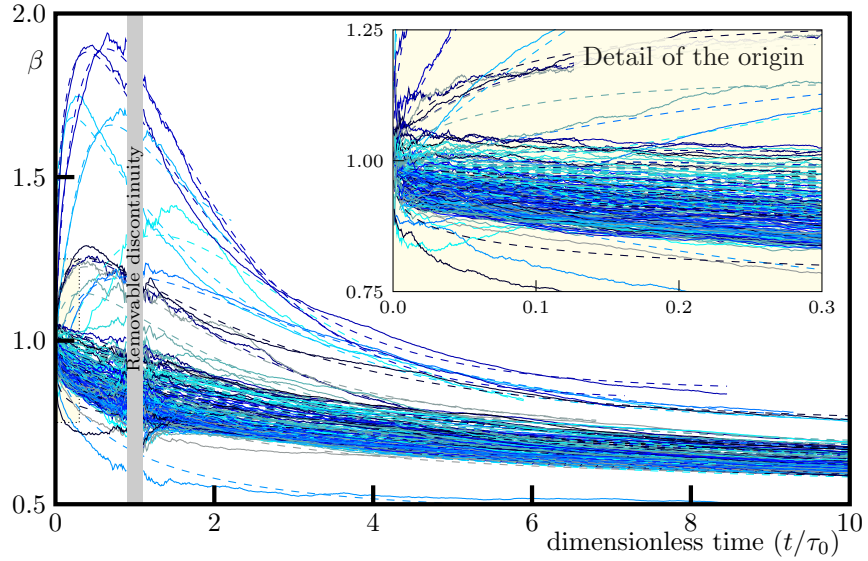

Figure 4.10: Measured exponents (continuous lines) and their approximations (dashed lines) as a function of dimensionless time. The results around 1 has been rejected due to large errors produced by a removable discontinuity (indeterminate form 0/0 of equation 4.88).

The first step to find the relation is to normalise both $[Y]$ and $[U]$. If N is the number of samples, then the mean value for each component is

$$\bar{y}_i = \mathbb{E}(y_i) = \frac{1}{N}\sum_{n=1}^{N} y_{i,n}. \tag{4.91}$$

The standard deviation σ can be also calculated as

$$\sigma^2(y_i) = \mathbb{E}\left[(y_{i,n} - \mathbb{E}(y_i))^2\right] = \frac{1}{N}\sum_{n=1}^{N}(y_{i,n} - \bar{y}_i)^2\,. \tag{4.92}$$

After calculating these two parameters the samples can be normalised as

$$\hat{y}_{i,n} = (y_{i,n} - \bar{y}_i)\,/\sigma(y_i). \tag{4.93}$$

Every normalised variable has mean value 0 and standard deviation 1. Something to take into consideration is the anomalous points. Here a sample is considered anomalous if one of the absolute values of the normalised components exceeds 3. When it happens the anomalous sample is rejected and the normalisation process is recomputed. This process is iterative and it finishes when no more samples are rejected.

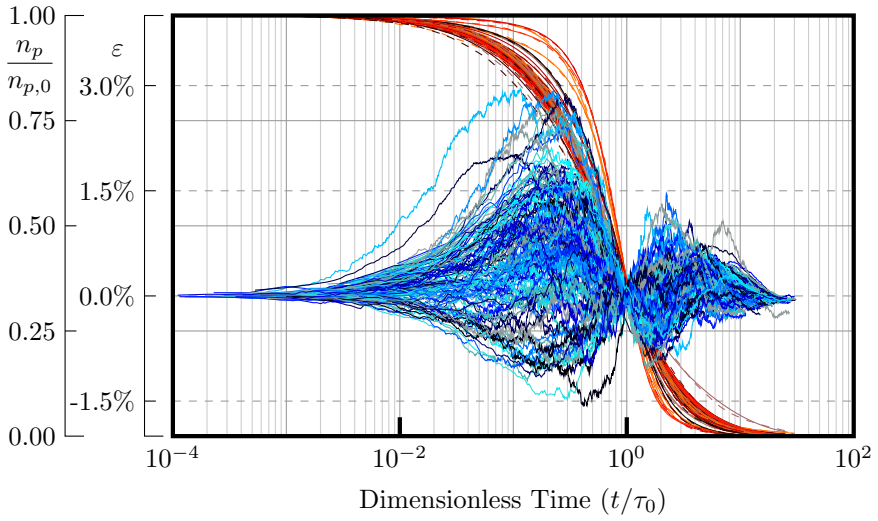

Figure 4.11: In red tones, evolution of the number of agglomerates with time (continuous line) and the approximation (dashed line). In blue tones, error of the approximation defined as the difference between the measured values and the expected values.

The next step is the orthogonalisation. Working with orthogonal variables is usually cheaper. The information extracted during the process can be used to detect over-fitting. In addition, when over-fitting is detected, the number of independent variables can be reduced, simplifying some of the calculations. About the physical parameters, they have been generated randomly using Lattice Hypercube. Orthogonality is one of the properties of the samples generated by this method, so no transformation is required for this dataset. On the contrary, the fitting parameters are not orthogonal. A set of orthogonal intermediate variables will replace them. The orthogonalisation process starts with the calculation of the correlation matrix $[\Sigma]$. Considering the dataset $[Y]$ the correlation matrix can be calculated as

$$[\Sigma] = [\hat{Y}]^T[\hat{Y}]. \tag{4.94}$$

Two sets of samples $[Y]_i$ and $[Y]_j$ are orthogonal between them if they are not correlated, $[\Sigma]_{ij} = 0$. In other words, the variables are orthogonal if their correlation matrix is diagonal. Consequently the orthogonalisation process consists on diagonalising the correlation matrix. This task can be done by eigendecomposition

$$[\Sigma] = [V][\Lambda][V]^{-1}. \tag{4.95}$$

$[\Lambda]$ is a diagonal matrix whose component λ_{ii} is the i-th larger eigenvalue of $[\Sigma]$ and $[V]$ is the matrix whose i-th column contains the eigenvector associated

with this i-th eigenvalue. If the eigenvectors are unitary $[V]^{-1} = [V]^T$. Premultiplying the equation 4.95 by $[V]^T$, postmultiplying by $[V]$ and replacing $[\Sigma]$ from equation 4.94 the expression

$$[V]^T[\hat{Y}]^T[\hat{Y}][V] = [\Lambda] \tag{4.96}$$

is returned. It can be observed how the diagonal matrix $[\Lambda]$ is the correlation matrix of $[Y_\perp] = [\hat{Y}][V]$ (see equation 4.94). If the correlation matrix is a diagonal matrix the variables are orthogonal, so $[Y_\perp]$ contains the orthogonal transformed samples of the fitting parameters.

In some situations the lowest eigenvalues could be some order of magnitude smaller than the greatest one (ill-conditioned correlation matrix). It is a sign of over-fitting. Under this conditions a truncated decomposition can approximate very well the original matrix. It consists on replacing the lower eigenvalues by 0. Every substitution reduces in one the number of intermediate variables. This reduction could be of great interest, specially if the number of variables is very large, when the curse of dimensionality becomes the limiting factor.

Here the truncation has been used to avoid over-fitting. The square of the deviation magnitude of the fitting parameters is the sum of all eigenvalues. It is desired to kept at least the 95% of the variability, so the sum of the survival eigenvalues over the sum of all of them has to be at least 0.95. Denoting the truncated transformation by the superscript ^{f}

$$\lambda^f_{\perp,i} = \begin{cases} \lambda_{\perp,i} & \text{if} \quad \sum_{j=1}^{i} \lambda_{\perp,j} \Big/ \sum_{j=1}^{N} \lambda_{\perp,j} \leq 0.95, \\ 0 & \text{if} \quad \sum_{j=1}^{i} \lambda_{\perp,j} \Big/ \sum_{j=1}^{N} \lambda_{\perp,j} > 0.95. \end{cases} \tag{4.97}$$

The eigenvalues have to be sorted in decreasing order. The eigenvectors associated to the deleted eigenvalues disappear (columns of $[V]$), so $[V^f]$ is not square anymore. The filtered correlation matrix $[\Lambda^f]$ is reduced to the diagonal matrix with contains only the kept eigenvalues. The resultant matrix of samples $[Y^f_\perp] = [\hat{Y}][V^f]$ contains less variables (lower number of columns).

Finally it is time to find the relation between the transformed sets of physical parameters and fitting factors. This relation is

$$[\hat{U}][F] = [Y^f_\perp] \tag{4.98}$$

Multiplying by $[\hat{U}]^T$ and taking advantage of its orthonormal properties the resulting expression is

$$[F] = [\hat{U}]^T[Y^f_\perp] \tag{4.99}$$

In other words, the function which relates both sets of samples is the cross-correlation.

At this time the deterministic part is well characterised. It is time to describe the stochastic contribution. This information has to be extracted from the residuals (or errors, if there are extra experiments). Consequently the first task to do is to calculate the residuals:

$$[R] = [\hat{Y}] - [\hat{U}][F][V^f]^T. \tag{4.100}$$

The linear model has been approximated by mean least squares, so the residuals are already centred. Only it is needed to scale by the standard deviation. Again, the normalised residual population $[\hat{R}]$ is not necessary described by a set of orthogonal variables. In a similar way to the work done with the fitting factors the correlation matrix is decomposed (see expression 4.95) in the diagonal matrix with the eigenvalues and the matrices with the eigenvectors. If some of the eigenvalues are very small the transformation can be truncated (see expression 4.97). As it has been seen before, each eigenvalue is the square of the standard deviation of the population for each orthogonal direction. The final model for the normalised variables including uncertainty is

$$\vec{\hat{y}} = \vec{u}_\perp[F][V^f_{\hat{Y}}]^T + \vec{\hat{\chi}}_\perp[\Lambda^f_{\hat{R}}]^{1/2}[V^f_{\hat{R}}]^T\mathrm{Diag}(\vec{\sigma}_R) \tag{4.101}$$

where $\vec{\hat{\chi}}_\perp$ is a vector of random samples extracted from a population whose mean value is equal to 0 and its standard deviation is equal to 1.

For the case of interest the number of independent fitting factors is four (with the 95% criterion). It means that the other two are dependent, linear combination of the others. The obtained eigenvalues of the correlation matrix are

$$[\Lambda_{\hat{Y}}] = \mathrm{diag}([\ 2.94 \quad 1.41 \quad 0.72 \quad 0.67 \quad 0.08 \quad 0.02\ ]). \tag{4.102}$$

The starting four are conserved. The truncated eigenvector matrix is

$$[V^f_{\hat{Y}}]^T = \begin{bmatrix} -0.3870 & -0.5196 & -0.5192 & -0.0165 & -0.1287 & -0.5421 \\ 0.2188 & -0.0790 & -0.2600 & 0.7079 & 0.6141 & 0.0011 \\ 0.7841 & -0.4610 & 0.1967 & 0.0170 & -0.2745 & -0.2416 \\ -0.1203 & 0.0490 & -0.1009 & 0.6217 & -0.7107 & 0.2853 \end{bmatrix} \tag{4.103}$$

The coefficient matrix for normalised parameters is

$$[F][V^f_{\hat{Y}}]^T = \begin{bmatrix} -0.7253 & 0.2203 & -0.1552 & -0.2720 & -0.1672 & 0.0513 \\ -0.0636 & 0.0029 & -0.0158 & -0.0039 & -0.0935 & 0.0075 \\ -0.0250 & 0.0479 & 0.0328 & -0.0177 & -0.0603 & 0.0481 \\ -0.5500 & -0.7054 & -0.6701 & -0.0079 & -0.4651 & -0.6843 \\ -0.1652 & 0.0240 & -0.0425 & -0.0273 & -0.1486 & 0.0143 \end{bmatrix}. \tag{4.104}$$

This matrix is the cross-correlation matrix[6]. It can be observed how the dominant factors are usually $\log_{10}(n)$ and $\log_{10}(\Delta E_{1/2})$. The concentration affects

[6] After considering negligible two of the six fitting factors this matrix is an approximation to the cross-correlation matrix instead of the cross-correlation matrix itself.

principally τ_0. The energy, when the probability to collide and agglomerate is the same, affects every variable except c_1. c_1 is practically uncorrelated with any variable. It means that this parameter is practically insensible to the inputs and its variability is part of the physics. In other words, this variable is chaotic and it has to be expressed by means of a PDF.

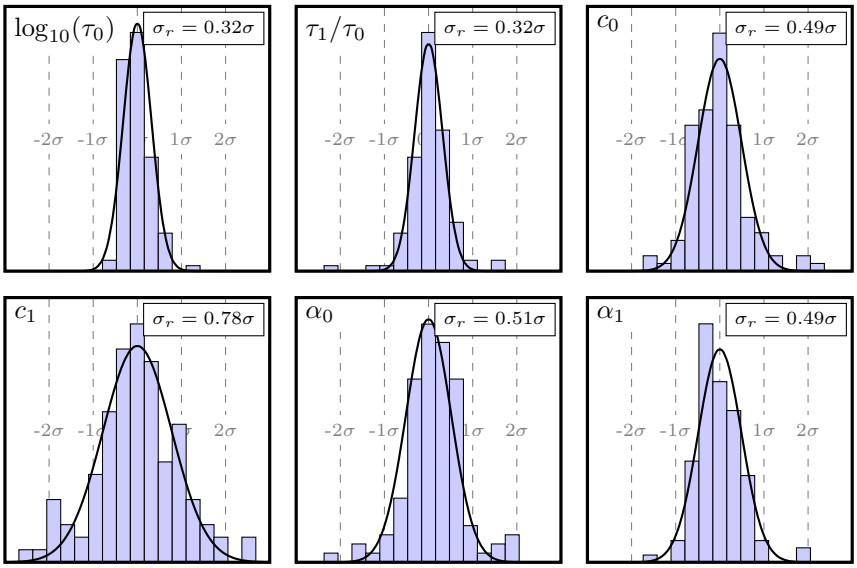

Figure 4.12: Histograms of the residuals for each variable and the best fitting normal distribution. The values of the deviations of the residuals σ_r are given as a function of the deviations of the measured populations σ.

Once the linear model is fitted it is time to characterise the uncertainty. The residual populations and the best fittings to the normal distribution are shown in figure 4.12. The orthogonalisation process returns the eigenvalues

$$[\Lambda_{\hat{R}}] = \mathrm{diag}([\ \ 2.15\mathrm{E}{-2} \quad 1.36\mathrm{E}{-2} \quad 8.26\mathrm{E}{-3} \quad 1.93\mathrm{E}{-3} \quad 1.26\mathrm{E}{-3} \quad 2.94\mathrm{E}{-4}\ \ ]). \tag{4.105}$$

The last two eigenvalues are small enough in comparison with the largest one to be neglected. The eigenvectors associated to the preserved eigenvalues are

$$[V^f_{\hat{R}}]^T = \begin{bmatrix} 0.42 & 0.54 & 0.47 & 0.09 & -0.04 & 0.55 \\ 0.04 & 0.02 & -0.38 & 0.69 & 0.58 & 0.21 \\ -0.63 & 0.32 & -0.23 & 0.29 & -0.55 & 0.26 \\ -0.62 & 0.29 & 0.33 & -0.29 & 0.59 & 0.00 \end{bmatrix} \tag{4.106}$$

After operating with the resultant matrices the linear approximation which relates the physical parameters with the fitting parameters is takes the form

$$\boldsymbol{y} = [\mathscr{M}_0]\boldsymbol{x} + [\mathscr{M}_1]\boldsymbol{\xi} + \boldsymbol{c}$$

$$\boldsymbol{y} = \begin{Bmatrix} \log_{10}(\tau_0) \\ \tau_1/\tau_0 \\ c_0 \\ c_1 \\ \alpha_0 \\ \alpha_1 \end{Bmatrix}, \boldsymbol{x} = \begin{Bmatrix} \log_{10}(\ n\ ) \\ \log_{10}(\ 1 - C_R^2\ ) \\ \log_{10}(\ \sigma_S\ ) \\ \log_{10}(\ \Delta E_{1/2}\ ) \\ \log_{10}(\ \Delta E_\sigma\ ) \end{Bmatrix}, \boldsymbol{\xi} = \begin{Bmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \end{Bmatrix}, \boldsymbol{c} = \begin{Bmatrix} -1.35 \\ 0.83 \\ 0.59 \\ 0.37 \\ 0.52 \\ 0.57 \end{Bmatrix},$$

$$[\mathscr{M}_0] = \begin{bmatrix} -1.02\text{E}{+0} & -2.94\text{E}{-2} & -3.97\text{E}{-2} & -1.99\text{E}{-1} & -1.69\text{E}{-1} \\ 2.99\text{E}{-2} & 1.28\text{E}{-4} & 7.33\text{E}{-3} & -2.46\text{E}{-2} & 2.36\text{E}{-3} \\ -1.22\text{E}{-2} & -4.06\text{E}{-4} & 2.91\text{E}{-3} & -1.35\text{E}{-2} & -2.42\text{E}{-3} \\ -2.76\text{E}{-2} & -1.31\text{E}{-4} & -2.03\text{E}{-3} & -2.07\text{E}{-4} & -2.01\text{E}{-3} \\ -1.55\text{E}{-2} & -2.84\text{E}{-3} & -6.30\text{E}{-3} & -1.11\text{E}{-2} & -9.99\text{E}{-3} \\ 1.40\text{E}{-2} & 6.73\text{E}{-4} & 1.48\text{E}{-2} & -4.80\text{E}{-2} & 2.84\text{E}{-3} \end{bmatrix},$$

$$[\mathscr{M}_1] = \begin{bmatrix} 7.81\text{E}{-2} & 5.67\text{E}{-3} & -7.30\text{E}{-2} & -3.46\text{E}{-2} \\ 1.50\text{E}{-2} & 3.50\text{E}{-4} & 5.39\text{E}{-3} & 2.38\text{E}{-3} \\ 1.02\text{E}{-2} & -6.62\text{E}{-3} & -3.06\text{E}{-3} & 2.14\text{E}{-3} \\ 4.06\text{E}{-3} & 2.39\text{E}{-2} & 7.79\text{E}{-3} & -3.78\text{E}{-3} \\ -1.17\text{E}{-3} & 1.27\text{E}{-2} & -9.35\text{E}{-3} & 4.83\text{E}{-3} \\ 3.44\text{E}{-2} & 1.02\text{E}{-2} & 1.02\text{E}{-2} & 2.45\text{E}{-5} \end{bmatrix}. \quad (4.107)$$

Figure 4.13 shows the simulated population and a generated population for the same physical parameters. The abscissa axis represents the expected value, the approximation returned by the linear model. The ordinates are the measured values and the random generated population resulting from expression 4.107, when the uncertainty is added to the expected value. The distributions considered to generate the stochastic contribution are normal with mean 0 and deviation 1.

Several observations are required to be taken into account about figure 4.13:

- Some slight non-linearities have been observed for τ_1 and α_1. The measured samples follows a curve instead of a straight line. This non-linearity is not big enough to invalidate the model. However the discrepancy between the expected values and the data has been included as part of the uncertainty. The non-linear effects, deterministic in origin, become stochastic in the model.

- c_0, α_0 and more specially c_1 present high uncertainty. The cloud covers large vertical distance (deterministic and stochastic) in comparison with the horizontal one (purely deterministic).

- The population of figure 4.13 has been created supposing normal distribution of the residuals. This assumption is not necessary true. More realistic populations can be generated when the PDF is obtained from

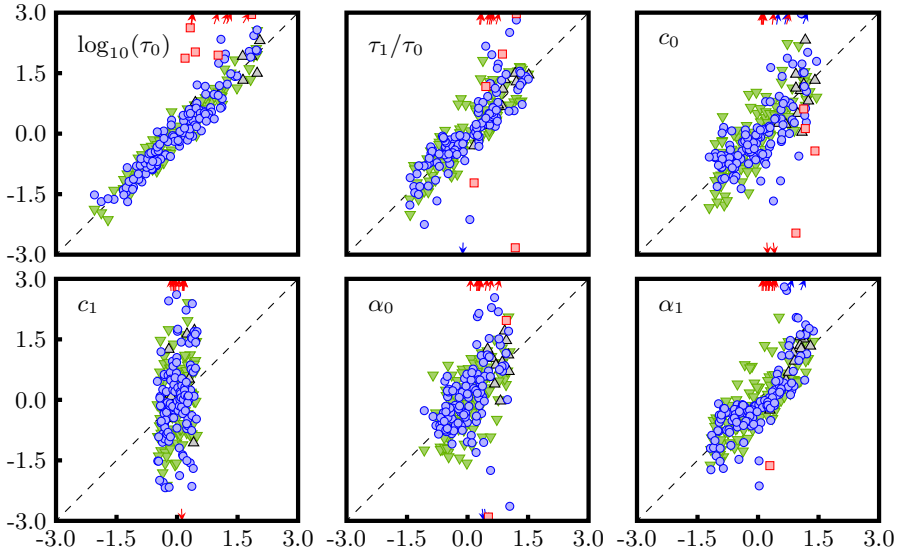

Figure 4.13: Measured population (blue circles) and generated population (inverted green triangles) vs. the expected value for each fitting parameter in normalised magnitudes. The measured anomalous points are squares in red and their approximations are represented by black triangles. The dashed line represents ideal fitting. The arrows point out samples located outside of the limits.

the histograms of the residuals. This option requires much more samples than the calculated for this study. One case when this hypothesis is not true is the population generated for c_0. This normal distribution underestimates the concentration at the middle in detriment of an overestimation at the edges.

4.3.5 Evolution of the number of agglomerates in space. Generating solid phase.

The objective of the simulation in space is to create porous media by the agglomeration of particles in ballistic regime. The resulting porous medium is expected to be equivalent to depositions of particles on the walls. The properties of the generated material can be characterised by fluid dynamics simulations. The macroscopic physical properties can be obtained by measuring mass flow rate and pressure drop when the fluid crosses through it. In the following lines it is explained how a porous media formed by agglomeration of

spheres is created in ballistic regime.

Simulation setup

It is wanted to create a block of porous media to characterise the properties of depositions of agglomerates on walls. Several aspects have to be taken into consideration.

- It is wanted to study only the effects of pure porous media material, so no solid wall can exist.
- The size of the sample has to be large enough to dilute the effects of anomalies.
- The thickness of the material has to be large enough to get uniform statistical properties.
- Impermeable lateral BCs could channelise the fluid. This effect is avoided by means of periodicity between lateral limits.

All of these aspects have to be accomplished to get an acceptable sample of the material. Periodicity is the most easy and intuitive condition. Two of the three directions have to be periodic. It means that every particle in the domain also exists in every period. A particle can collide with other agglomerate in the domain but also with the agglomerate in the neighbour period.

Periodicity also fixes the moment when a particle becomes solid phase. Because a particle exists multiple times, one per period, it can collide with itself. When it happens the particle becomes a new solid. A peculiarity of the solid with respect to a particle is rotation. A particle can rotate. The solid phase does not. It is infinite. Consequently it has only three degrees of freedom for translation.

Once a particle becomes a solid its precursors are moved to the first period for every time step. This can be done because the inertia matrix is not needed anymore. This strategy reduces the radius of the circumscribed sphere and as a direct consequence improves the performance. The physics of the solid phase continues to be equal with respect to collisions with other solid phases and agglomerates, but with infinite inertia matrix (no rotation). When a particle impinges the solid phase it impinges in every period. So the mass to take into consideration to calculate the acceleration of the solid phase is the mass contained in the first period.

With respect to the other BCs, one of them communicates with a reservoir of precursors (precursor inlet) and the other is just an output. The particle

inlet injects precursors from an equilibrium distribution with mean velocity 0 (see section 3.2.3). The agglomerates which go beyond the outlet leave the domain. This fact is closely related with the minimum length of the domain required for the simulation. If the domain is too short, every particle will be lost before forming any solid phase. Consequently very large domains are required for the simulations.

But this is not the single problem related with the movement of particles and the solid phase. The particles are only injected from the inlet. The solid phase is always impinged on the same side. Due to the momentum exchange the solid phase becomes faster and faster with every collision. At the end it is one of the fastest objects in the simulation and leave the domain too soon, being too small to be a candidate for porous media simulation. This problem is solved by imposing a drag force proportional to the velocity for every solid phase $\vec{F} = C_D u_\infty$. It slows down the object till compensate the momentum exchange produced by the impacts of agglomerates against it. The drag coefficient has to be large enough to keep the solid phase in the domain but not too large to grow till the inlet. An approximation of the terminal velocity u_∞ can be estimated from equation

$$\Delta A m_p n \int_0^\infty \mathrm{pdf}(c_x)\,(c_x - u_\infty)\,dc_x = C_D u_\infty \tag{4.108}$$

where ΔA is the front area of the domain (size of the inlet), m_p the expected mass of the precursors and n the concentration. The left hand side is the relative momentum inlet with respect to the solid phase. The right hand side is the momentum change due to drag force. Both of them are compensated (no acceleration term) in the previous equation.

Simulations

A unique simulation has been run. The parameters which controls the physical behaviour are listed in table 4.2.

A peculiarity of fractal geometries is self-similarity between different scales. The resulting porous media is one this geometries. The auto-correlation of this type of structures is never zero. There is always a coherent structure as big as the domain independently of the detail level. The replacement of the most energetic agglomerations by collisions has as objective to reduce this effects at large scales and avoid the existence of non-physical structures, when the resistance of the material is not strong enough to keep the form of biggest structures without collapsing. 200 radii have been considered big enough to dilute the 3D effects of the precursors. Pressure drop through the porous media is dominated by the lowest scales.

Magnitude	Symbol	value	Units
Domain size	Ω	$200 \times 200 \times 9600$	$r_{\text{ref}} \times r_{\text{ref}} \times r_{\text{ref}}$
Time step	Δt	0.05	r_{ref}/σ_u
Density of the material	ρ	1.0	ρ
Energy lost for impacts	ΔE	5%	-
Reference radius of precursors	$\log_{10}(r_{\text{ref}})$	0.0	-
Deviation of the log-normal distribution in radius	$\log_{10}(\sigma_r)$	0.13	-
Mean speed of the reservoir	$\bar{u}$	0.0	σ_u
Speed deviation at the reservoir	σ_u	1.0	σ_u
Concentration	n	$5{\times}10^{-3}$	particles r_{ref}^{-3}
Damping factor for solid phase	C_D	5.0	$\rho\sigma_u r_{\text{ref}}^2$

Table 4.2: Setup to simulate the generation of solid phase.

To give time enough to the solid phase to be created the domain is 9600 radii long in streamwise direction. A rich concentration of precursors has been considered for the same reason. The damping factor has been selected to keep the solid phase in the domain at the same time that it grows up. The energy loss during the collision is very low, 5%. It differentiates the simulation from simple ballistic growing by addition, when every impact fuses the colliding agglomerates.

The simulation has been run in the work-station described before for few days. At the end a large solid phase formed by more than half million particles was created. The results and the evolution of the simulation are detailed in next section.

Results

The simulation can be divided in three different stages. At the beginning there is no solid phase. The agglomerates grows up with the distance from the inlet. After a while some of the agglomerates becomes very large and they start to collide with their periodic copies. At this points several solid phases coexist. The most lagging solid phases receive all the impacts of incoming agglomerates. The collisions accelerate them. On the contrary the most advanced agglomerates are not pushed by any impact and only the damping factor acts on them. As consequence they slow down. At the end due to the difference in speed the phases hit each other to be fused into a bigger one. This is the third stage of the simulation, when a big solid phase is continuously growing and it occupies most of the domain. The three stages of the simulation (starting part

of the domain) are represented in figure 4.14 for the sake of understanding.

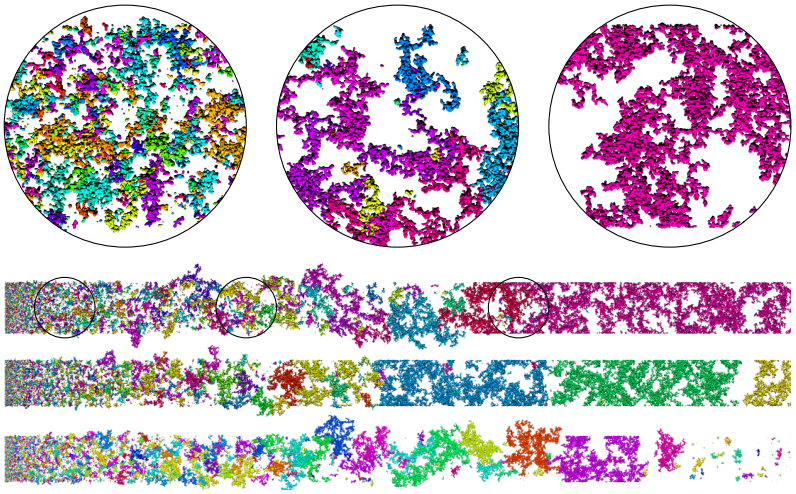

Figure 4.14: Circles: detail of the incoming particles (left), agglomerates (middle) and solid phase (right). Bars: simulation after 9 (bottom), 16 (middle) and 32.55 (top) time units. The domain is cut and zoomed to show more details.

The resultant material has large holes and clusters. The pressure drop is expected to be dominated by the thinnest of these holes.

4.3.6 Conclusion

A simulator for ballistic agglomeration processes has been created. Both collision and agglomeration can happen after contact. A probabilistic discriminant has been proposed to decide when two impinging agglomerates collide or when they fuse. The returned probability to collide or agglomerate depends on the energy absorption required for the fusion process. The agglomeration is less probable when the energy to be absorbed is large.

The size of the agglomerates covers several orders or magnitude, from the agglomerate formed by a single precursor to large agglomerates whose size is comparable to the size of the domain. The computational parallelisation becomes very difficult under these conditions. Two different divide and conquer strategies have been applied together. In first place the whole domain is divided in blocks recursively for each time step. The search of collisions starts when the number of agglomerates contained in each block is lower than a threshold. Two agglomerates are candidates to collide if their regions of

influence (circumscribed spheres) intersect each other. An agglomerate can be formed by thousands of precursors. Trying to locate the contact point precursor by precursor checking all the combinations is prohibitive. For this reason the precursors of the largest agglomerate which are outside of the area of influence of the smallest agglomerate are rejected. A second divide an conquer algorithm is then applied, reducing recursively the number of combinations. Each combination is checked when the number of required operations is under a threshold.

Two sets of simulations have been lunched. One to see the evolution in space, one to see the evolution in time. From the resultant populations the fractal dimension D_f has been calculated. It relates the gyration radius of the agglomerate R_g over the gyration radius of the typical precursor $R_{g,0}$ with the number of precursors n_p as $n_p = k_p(R_g/R_{g,0})^{D_f}$. After fitting the equation the obtained fractal dimension is 1.90 and the logarithm of premultiplicative constant $\log_{10}(k_p)$ is $-0.54^{+0.22}_{-0.24}$ for 95% of confidence interval.

The simulations of the evolution in time demonstrate how the agglomeration process is dominated by the initial concentration of precursors and the capability of the material to absorb the energy after an impact. A modified decaiment model has been proposed to explain the evolution of the concentration of agglomerates with time. The model includes six fitting parameters on which only four are linearly independent. The relation between the physical magnitudes, which includes properties of the material and initial conditions, and the fitting parameters have been established. In addition to the deterministic model, four stochastic variables have been introduced to explain the large natural variability observed in the results of the simulations.

A second experiment creates solid phase from deposition of agglomerates. An agglomerate is considered solid phase when it collides with its copy of the neighbour period. At the beginning multiple solid phases exist. The solid phases closest to the inlet receive most of the impacts from the incoming agglomerates and accelerate. Because the lagging solid phases shield the most advanced against collisions the furthest solid phases slow down. At the end the solid phases collide among them creating a large single one.

The resultant structure keeps the fractal behaviour of the agglomerates (figure 4.7). The largest structures are always of the same order than the domain independently of the scale. Consequently the surrounding flow field is expected to be also correlated. No domain size independent behaviour can be achieved with this model. It is needed structural analysis.

The solid phase can collapse when it is subjected to high structural loads. The collisions can not only fuse the colliding agglomerates, they can also break them into smaller pieces. The largest holes disappear creating more compact

solid phases with free moving agglomerates in the interior. This task is very hard to be developed so it is included in possible future works.

Bibliography

[1] Sorensen, C.M. The mobility of fractal aggregates: A review. *Aerosol Science and Technology*, 45(7):(2011), 755–769. Cited By 105.

[2] Dietzel, M.; Ernst, M.; Sommerfeld, M. Application of the lattice-boltzmann method for particle-laden flows: Point-particles and fully resolved particles. *FLOW TURBULENCE AND COMBUSTION*, 97(2):(2016), 539–570.

[3] Dietzel, M.; Sommerfeld, M. Numerical calculation of flow resistance for agglomerates with different morphology by the lattice-boltzmann method. *POWDER TECHNOLOGY*, 250:(2013), 122–137.

[4] Einstein, A. Investigations on the theory of the brownian movement, edited by r. *Fürth, Methuen, London*.

[5] Wang, G.M.; Sorensen, C.M. Diffusive mobility of fractal aggregates over the entire knudsen number range. *Phys. Rev. E*, 60(3):(1999), 3036–3044.

[6] Eggersdorfer, M.L.; Pratsinis, S.E. The structure of agglomerates consisting of polydisperse particles. *Aerosol Science and Technology*, 46(3):(2012), 347–353. Cited By 32.

[7] Eggersdorfer, M.L.; Pratsinis, S.E. Agglomerates and aggregates of nanoparticles made in the gas phase. *Advanced Powder Technology*, 25(1):(2014), 71–90. Cited By 28.

[8] Goudeli, E.; Eggersdorfer, M.L.; Pratsinis, S.E. Coagulation-agglomeration of fractal-like particles: Structure and self-preserving size distribution. *LANGMUIR*, 31(4):(2015), 1320–1327.

[9] Goudeli, E. and Eggersdorfer, M. L. and Pratsinis, S. E. title. Coagulation of agglomerates consisting of polydisperse primary particles. *LANGMUIR*, 32(36):(2016), 9276–9285. PMID: 27536889.

[10] Morgan, N.; Kraft, M.; Balthasar, M.; Wong, D.; Frenklach, M.; Mitchell, P. Numerical simulations of soot aggregation in premixed laminar flames. *Proceedings of the Combustion Institute*, 31(1):(2007), 693–700.

[11] Dekkers, P.J.; Friedlander, S.K. The self-preserving size distribution theory. *Journal of Colloid and Interface Science*, 248(2):(2002), 295–305.

[12] Meakin, P. A historical introduction to computer models for fractal aggregates. *Journal of Sol-Gel Science and Technology*, 15(2):(1999), 97–117.

Chapter 5

Filter simulations

There are several antecedents of filter simulations at micro-scale level. Sukop *et al.* [1] calculate porous media for multiphase fluids. They compare the simulation with real X-ray measurements for the same sample. As usual, LBM is used to calculate the flow field. Hayashi and Kubo [2], in addition to simulations at macro-scale level, compute the flow field through bed of spheres. The procedure followed by Hayashi and Kubo is similar to the procedure described on this thesis. At the beginning they create the geometry of the micro-structure. It is a bed of spheres that mimics a filter made of silicon carbide. The flow field is computed using LBM. Once the flow field is solved they compute the trajectory of particles and deposition. Bernsdorf, Brenner and Durst [3] also calculate the flow field through a a bed of spheres. They validate the numerical results with experiments. Rong, Dong and Yu [4] calculate bed of spheres too. Unlike the previous case, the spheres are of multiple sizes. They follow a distribution. Rong, Zhou and Yu [5] extend the study to uniform population of ellipsoids. Yamamoto and his team [6, 7, 8, 9, 10, 11, 12] also solve the flow field using LBM. However, they obtain the micro-structure by X-ray tomography. In addition to the flow field and depositions they also simulate regeneration. Stewart *et al.* [13] also simulate the entire filtering process. The geometry is also obtained by tomography. The simulated filter is made of acicular mullite. The objective of the work is to predict the filter behaviour: pressure drop during filter loading and deep of soot penetration in the substrate. Stewart *et al.* also study the effects of tangential velocity upstream the porous wall.

5.1 Simulation setup

There are several parameters to take into account before calculating any simulation. These parameters are related with the domain size and the resolution of the discretisation. To simulate correctly porous media the domain has to be large enough to be statistically representative and to dilute the effects of the BCs With respect to the resolution of the discretisation, it has to be fine enough to capture the details of the geometry. The first part of this chapter is dedicated to explain the procedure to do a quality filter simulation. The procedure can be used to validate the quality of the simulation or to prepare the previous setup. The geometry is created by procedural generation. It mimics the micro-structure of the wall of a Diesel filter made of acicular mullite.

5.1.1 Geometry generation

The geometry to be calculated mimics a filter made of acicular mullite. The micro-structure of this material is composed of very long, interconnected crystals (like needles). They form randomly-distributed radial aggregates. The origin, where the crystals grow from, will be called seed of the aggregate from now. Furthermore there are not two equal crystals. Their length, thickness, orientation, location... are random.

Accordingly to the previous description a population of crystals has been created following the process described in in section 2.4.1. It is assumed that the crystals have been synthesised from a block of material whose thickness is 200μm. That is, the porous media is 200μm thick and located between the y coordinates -100μm and 100μm. That means that every seed will be contained in this interval. A uniform random distribution has been considered for every direction. The concentration of aggregates per mm^3 is 50,000.

Once the origin of every aggregate is located, the crystals have to be created. A normalised crystal is scaled to obtain its final length and thickness. For the sake of simplicity random uniform distributions have been considered for both the length and the thickness. The minimum length for the crystals is 10μm and the maximum 100μm. To avoid the creation of very anomalous shaped crystals the thickness is not generated directly. The aspect ratio defined as the thickness over the length is used instead. It is limited from 0.01 to 0.05 times the length. The number of crystals for each aggregate may also change. Samples from a uniform distribution between 1 and 64 have been used to specify this parameter.

Because the crystals do not grow by the same length in each of the two directions their centres diverge from the seed of the aggregate. The distance

between them is called eccentricity. This effect is modelled again by a uniform distribution. Same as for the thickness of the crystals, the eccentricity has been modelled relative to the length. In this case the value is limited from -0.7 to 0.7 times the length.

Once the crystal is scaled and off-centred it has to be oriented. Before any modification the principal axis follows the X axis. To achieve the final orientation two consecutive rotations are applied. Firstly, the elevation, a rotation around the Y axis bounded by the interval $[-\pi/2, \pi/2]$. Secondly, the azimuth, a rotation around the Z axis bounded by the interval $[-\pi, \pi]$. A uniform distribution per steradian can be achieved by a uniform distribution for the azimuth. However, the same type of distribution can not be used for the elevation, otherwise an over-concentration appears at the zenith. To compensate this effect a uniform distribution bounded by the interval [-1,1] is created to be modified. Once the population is generated the arcsin() operator is applied to each random value, resulting in the elevation sample.

5.1.2 Detection of the smallest scales: fractal dimension of the interface

When the domain is discretised the resolution of the discretisation has to be fine enough to capture the smallest details. In porous media analysis the smallest details are provided by the interface between solid and fluid. In other fields like turbulence the smallest scales are the fluid structures instead of the geometry of the BCs. One way to measure the level of detail is the fractal dimension. The interface is a surface so its fractal dimension is two [14]. Higher values appear when the discretisation is too coarse and the elements include entire small structures.

One of the more widely extended techniques to measure the fractal dimension is the box-counting algorithm. Block, Vonbloh and Schellnhuber [15] propose this efficient method to calculate the fractal dimension numerically. It returns the Minkowski–Bouligand dimension $\mathscr{D}_{MB}$. The technique consists on dividing the domain into cubic cells with h long edges. After dividing the domain it is counted how many cells N contain part of the interface. The Minkowski–Bouligand fractal dimension is defined as the derivative of the logarithm with respect to the logarithm of the scale

$$\mathscr{D}_{MB} = \lim_{h \longrightarrow 0} -\frac{\ln(N(h))}{\ln(h)}. \tag{5.1}$$

Applying the l'Hôpital rule, the limit can be approximated by the derivatives

respect to h of the numerator and the denominator

$$\mathscr{D}_{MB} = \lim_{h \longrightarrow 0} -\frac{\partial_h \ln(N(h))}{\partial_h \ln(h)}. \tag{5.2}$$

The derivative of a generic function $f(h)$ respect to h can be approximated numerically as $\partial_h f(h) = \Delta h^{-1}[f(h_{i+1}) - f(h_i)]$. Consequently the Minkowski dimension can also be approached numerically for each scale h_i as $\mathscr{D}^*_{MB}(h_i)$. Then the expression (5.2) becomes

$$\mathscr{D}^*_{MB}(h_i) = -\frac{\ln(N(h_{i+1})) - \ln(N(h_i))}{\ln(h_{i+1}) - \ln(h_i)}. \tag{5.3}$$

The approximation tends to the original definition when the scale h_i tends to be zero

$$\mathscr{D}_{MB} = \lim_{h_i \longrightarrow 0} \mathscr{D}^*_{MB}(h_i). \tag{5.4}$$

This is consistent with the limit definition of expression (5.2). An approach is considered good when the Minkowski dimensions is insensitive to the scale h_{i+1}. To take advantage from the logarithms the sequence of h_i is considered geometric. The common ratio of the progression r is the inverse of a positive integer number R, called the refinement factor, so $r = 1/R$ and $h_i = rh_{i-1} = r^i h_0$. Then the equation (5.3) is simplified to

$$\mathscr{D}^*_{MB}(h_i) = -\frac{\ln[N(h_i)/N(h_{i-1})]}{\ln(r)}. \tag{5.5}$$

The calculation of the Minkowski dimension is as follows

1. The domain is divided into cubic cells whose sides are h_0 long.
2. Some cells contain the fluid phase, some cells contain the solid phase and the rest contain both the solid and the fluid phase. The number of cells of the latter type is counted getting $N(h_0)$.
3. The starting cell size h_0 is divided by an integer number R, obtaining h_1.
4. The cells where the fluid and the solid phases coexist are divided into R^3 cubes.
5. Again, some of the new cells contain the solid phase, others contain the fluid phase and the rest contain the interface. The number of cells which contain the latter type gets $N(h_1)$.
6. The first approximation to the fractal dimension $\mathscr{D}^*_{MB}(h_1)$ can be obtained from h_0, h_1, $N(h_0)$ and $N(h_1)$ applying the equation (5.5).

7. Applying the points 2 to 6 over the obtained result we can get $\mathscr{D}^*_{MB}(h_2)$, $\mathscr{D}^*_{MB}(h_3)$ and so on. The process is finished if $\mathscr{D}^*_{MB}(h_i) \simeq \mathscr{D}^*_{MB}(h_{i+1})$. Then it is considered $\mathscr{D}^*_{MB}(h_i) \simeq \mathscr{D}_{MB}$.

Operating on volumes the value of the fractal dimension is bounded $\mathscr{D}^*_{MB}(h_i) \in [0, 3]$.

This methodology is applied to measure the fractal dimension of the crystal surface. This study begins at a single cell that takes up the whole domain. Each cell is 11 times recursively divided into 8 cells to create the octree mesh which covers the entire interface with a refinement factor $R = 2$. It means that the starting cell, the unique one in level 0, contains 2^{33} cells of level 11.

Statistically speaking, a single cell is not representative at all. A large number of samples are needed to obtain a good approximation of $\mathscr{D}^*_{MB}$. The bigger the number of samples, the better the approximation. It is then important to know where is the limit when the number of samples starts to be representative. A self-similar geometry with known fractal dimension has to be analysed with the same procedure than the studied case for validation. The simplest case of self-similar surface is just a plane. It always looks like equal independently to the scale and its dimension is 2. Consequently a random-oriented plane was inserted into the octree mesh. The results of the preliminary study can be seen at figure 5.1 as one of the reference cases. Three refinement levels (2^9 cells) are needed to obtain a good approximation to the real value of the dimension. Any result coming from a lower level is not reliable. The analysis of the chaotic geometry has to be done taking into account this limitations.

The interface is a surface, that is, its dimensionality is two. As consequence the expected fractal dimension gets this value for high refinements. Therefore, behaviour and structures of larger scales are analysed in this section. Depending on the crystal density the geometry can either be analogised to a bunch of randomly distributed needles (low density of crystals) or a porous media where the fluid phase is delimited by the gaps between the crystals (high density of crystals). This is why several concentrations of structures have been checked. Concretely 2, 5, 10, 20 and 50 thousand structures per unit volume (mm^3). The figure 5.1 shows the results of this study. Depending on the scale and the concentration different values for each dimension can be observed.

1. **Lattice spacing grater than the porous media thickness.** It has been simulated a film of porous media where its thickness is lower than its extension. When the cell size is much larger than the layer thickness the solid phase is meshed as a plane. The lattice can not capture any detail of the micro-structure. As a consequence for a scale of the same order or larger than the porous media thickness the expected value of the dimension is 2, the dimension of the wall. This value can not be seen

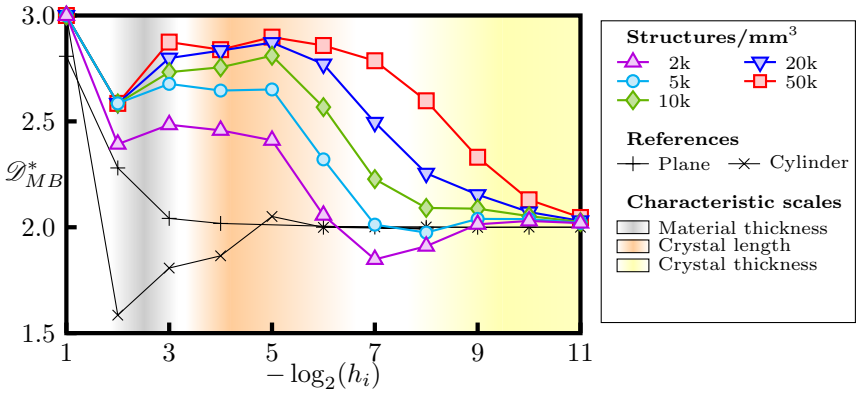

Figure 5.1: Fractal dimensions $\mathscr{D}^*_{MB}$ for each scale h_i for several concentration of acicular structures.

at figure 5.1 because the number of cells larger than the wall thickness is not large enough to be representative. There is a single cell of this size and it always contains the solid phase.

2. **Lattice spacing of the same order than acicular structures.** The porous media contains several cells across its thickness. However the cell sizes are too big to capture any detail. As consequence, specially for very dense micro-structures, a lot of cells enclose part of the interface. It means that most of the cells are divided. If it happens the dimension increases abruptly till values close to 3 for very dense geometries.

3. **Lattice spacing of the same order than the crystal thickness.** The results of $\mathscr{D}^*_{MB}$ for this scales are different depending on whether the wall is almost empty or filled with solid phase. This is related with the aspect ratio of the crystals, which is very high (length much larger than thickness). As consequence a single crystal could be seen as a one-dimensional object if the reference scale is larger than its thickness. One example of this can be seen at figure 5.1 where one of the reference cases is a cylinder. This is the reason why $\mathscr{D}^*_{MB}$ can take values lower than two when the density of crystals is very low. In the other hand, when the porous media is very dense, most of the interstices are too thin to be capture by the cell size. As consequence most of the contained volume is interpreted as interface. That is the reason why denser structures need stronger refinements to achieve value 2 in its dimension.

4. **Lattice spacing much lower than the crystal thickness.** No more details exist for lower scales than the crystal thickness. When the lattice spacing decreases beyond this value, the dimension shows what the solid-fluid interface really is, a surface. That is the reason why the expected

dimension value is something around 2 and why it has been chosen as the criteria for the upper bound of the minimum lattice spacing. The little discrepancies between the obtained value and the theoretical one is the incapability of the refinement to reproduce each interstice. In spite of this inconvenience the fluid flow through the unsolved holes is negligible as it could be observed during the mesh independent study.

5.1.3 Grid dependence study

To estimate the effects of the spatial resolution on the discretisation error, a grid study has been conducted. The original cell size of 2^{-6}mm is halved four times down to 2^{-10}mm. To avoid using four times as many time steps with every increase of resolution acoustic scaling and adaptation of the relaxation times λ_e and λ_o are applied as depicted in Table 5.1. This way, the number of time steps is doubled resulting in an increase of workload by a factor of 16 for every increment of resolution. In light of the results (cf. Fig. 5.2) the use of a resolution of 2^{-9}mm is considered to be a good compromise between accuracy and computational complexity.

Magnitude	**Case 0**	**Case 1**	**Case 2**	**Case 3**	**Case 4**
dx [mm]	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
dt [s]	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}
time steps	4,000	8,000	16,000	32,000	64,000
λ_e	1.902	1.814	1.660	1.418	1.099
λ_o	0.0976	0.186	0.340	0.582	0.901
flow rate [l/s]	0.350	0.157	0.116	0.099	0.094
rel. deviation	272.4 %	67.3 %	22.9 %	5.7 %	--

Table 5.1: Parameters which change during the grid study and the resulting flow rates. All Parameters refer to the finest grid level. The relative deviation states the difference of the flow rate result to the solution of the finest grid used.

5.1.4 Cross-correlations and autocorrelations for the microstructure and the velocity

The cyclic cross-correlations and cyclic autocorrelations are calculated over the periodic solutions along the periodic directions X and Z. The solution is smoothed by including every plane XZ which contains solid phase. For each Y, the FFT is applied on the fluid phase field and on every velocity component

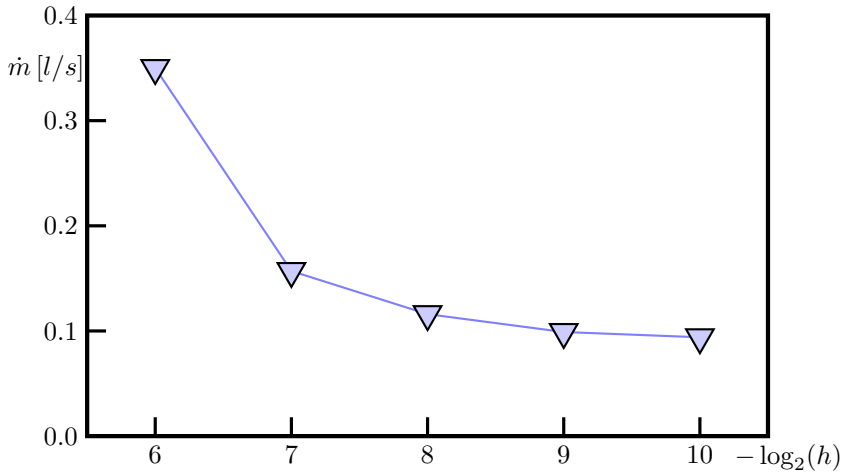

Figure 5.2: Grid study evaluating the flow rate through the porous medium using different spatial resolutions.

to obtain the spectra $f(x, y, z)$, getting one spectrum per plane and variable $\hat{f}(\kappa_x, y, \kappa_z)$ (f denotes a generic field). The cross-correlation of two fields f and g is calculated easily applying the equation 2.7 getting $\widehat{(f \star g)}(\kappa_x, y, \kappa_z)$. Then each spectrum is integrated along the Y direction getting a single spectrum per variable

$$\widehat{(f \star g)}(\kappa_x, \kappa_z) = \int_{-\infty}^{\infty} \widehat{(f \star g)}(\kappa_x, y, \kappa_z)\, dy. \tag{5.6}$$

Before any transformation of the correlations from the spectra to the physical domain the mean value is subtracted. That is, the first component for every variable with wave numbers κ_x=0 and κ_z=0 is assigned the value 0, $\widehat{(f \star g)}(0, 0) = 0$. Then the cross-correlations and autocorrelations are recovered in the physical domain $(f \star g)(\Delta x, \Delta z)$ applying the IFFT. However, they are not normalised. The inverse of the square root of the autocorrelations at zero displacement $(f \star g)^{1/2}$ is used as a normalising factor to get $\overline{(f \star g)}(\Delta x, \Delta z)$. Then the values are bounded by the interval [-1, 1].

The minimum domain size has to contain at least one of the largest chaotic identities. One of this identities includes both the phase field and also the affected surrounding flow field. Consequently, before any simulation a good estimation about the size of a single structure is required. Every identity correlates with itself, so it is expected to see strong correlations of the phase field for small displacements. If a clear pattern for a displacement exists in the autocorrelation, then there is an overlap of the structure with itself. It means

that the structure is at least as large as this displacement. On the contrary, if there is not a clear pattern and the solution for the autocorrelation of the phase field is clearly noisy, the displacement is larger than the identity. The domain is large enough then, to contain at least one structure. These autocorrelations can be calculated before any simulation and they are very fast to do. However, it is a necessary but not sufficient condition. The effects of the solid phase are extended to the surrounding area of the flow field. Consequently, the displacements for noisy autocorrelations of the flow field are expected to be larger. This validation has to be checked after the simulation as a post-processing task.

An exemplary case was run following the procedure described above in which the flow field through a porous wall is analysed. The domain is $0.5 \times 1 \times 0.5$mm long and the minimum cell size is 2^{-11}mm. The wall is supposed to be infinite but it has been replaced by a periodic approximation. It is extended along x and z directions and it occupies the centre of the domain in y direction. The analysed variables are the phase field and each component of the velocity field. Figure 5.3 shows the normalised results. Taking into account the isotropy of the porous medium the correlations with Z are not plotted. Those results are equivalent to the X correlations but rotated by 90 degrees.

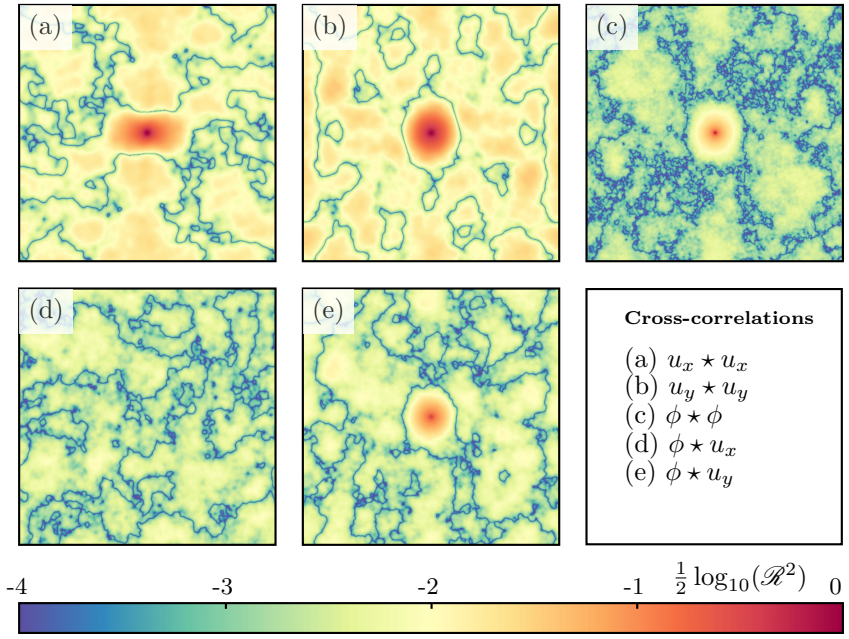

Figure 5.3: Normalised autocorrelations and cross-correlations of the phase and the velocity components. Because the porous medium is isotropic the z component is not shown (similar to x but turned ninety degrees).

As expected, a correlated region can be observed at the middle for all the autocorrelations. It is a circle centred in the origin for the phase field and u_y, and an ellipse in the case of u_x (and u_z). This also provides information about the characteristics of the geometry and the flow field. A circle indicates isotropy whereas an ellipse suggests a preference direction. Its size before the noisy solution reflects the range affected by a single structure. Then it can be concluded that the identity for the geometry is around 40μm long for all directions, which is the same order of magnitude than half of the length of a single crystal. Something similar happens with the autocorrelation of u_y. The range is approximately circular, again, meaning that u_y has always the same influence on itself independent of the direction X or Z. However, the affected area is much larger. The range is around 100μm, the maximum crystal length. It demonstrates how the fluid structures surround the solid structure. With respect to u_x and u_z, they are not isotropic. The major radius of the ellipse created by the correlation, is aligned along the velocity components. That means that the velocity of one point induces the same velocity in the neighbourhood. The maximum range is again 100μm.

The cross-correlations are also interesting to analyse. Especially important are the cross-correlations between the phase field and the velocity. They can show how the solid phase influences the velocity. It can be seen how the phase field and the perpendicular component y are correlated till displacements of 50μm. It is more or less the same radius of influence than the solid phase autocorrelation. However, no pattern can be observed when the solid phase is correlated with the velocity cross-components. As conclusion, the solid phase has no preference to induce velocity in any perpendicular direction.

The domain extension is appropriate for the simulation. It is more than twice the most restrictive criterion, the autocorrelation of u_y.

5.1.5 Influence of the peripheral boundary conditions

A simulation can be very sensible to the BCs (BCs). A good selection can make the difference between an accurate and an unrealistic solution. Every BC approximates the relation between the computational domain and the outer region. The better the approximation, the more accurate the solution. Porous media are no exception. The goal is the simulation of flow crossing a porous wall when it is subjected to a pressure gradient. The wall is supposed to be infinite across the XZ plane for $y = 0$. The thickness where the seed points are located is 200μm, between y=-100μm and y=100μm. For this case the correct BCs are fixed pressure at $y = -\infty$ and $y = \infty$. For the sake of simplicity of BCs an approximation is used instead. The fixed pressure BCs are moved closer to y=-500μm and y=500μm as is shown in figure (5.4). Of course some

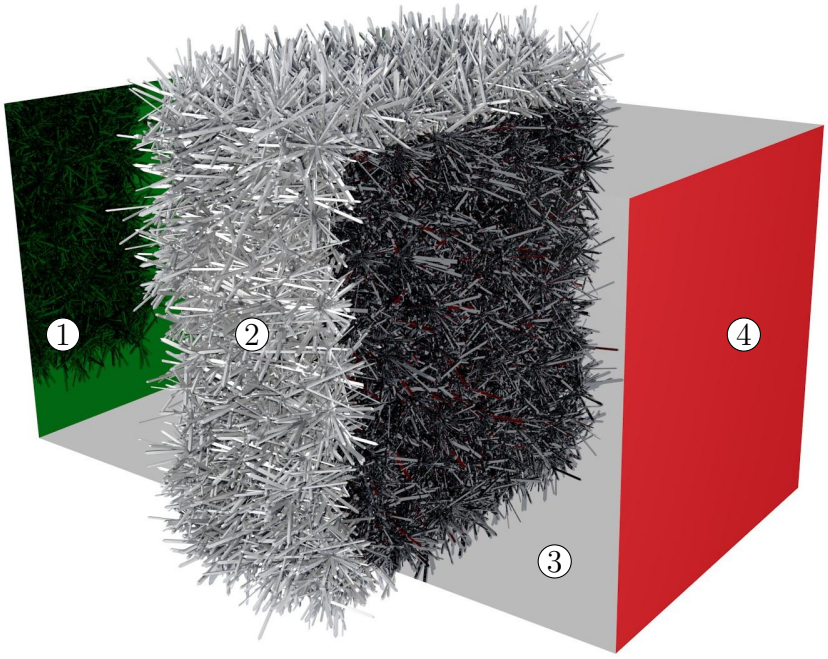

Figure 5.4: Boundary conditions for porous media simulations: (1) pressure or velocity inlet, (2) wall, (3) periodicity or symmetry, (4) pressure or velocity outlet.

little perturbations are not taken into account at these locations, but they are small enough to be neglected.

Something similar happens in X and Z directions. The real wall is infinite. Obviously an infinite wall can not be simulated so it is replaced by a sample of it. There is only one exception: the periodic domain. In this case the periodic BCs are exact, so the simulation of the sample coincides with the simulation of the infinite domain. When the wall has no periodic pattern then other BCs are needed to close the sample. Because the flux has to be conserved across the wall the total flux across the BCs has to be zero. The simplest and one of the most widely used BC is mirror or slip-wall. Contrary to the previous case, it produces strong perturbations at the surrounding area because this behaviour is unnatural across the wall. The flow at this region is forced to be tangential. If the fluid is artificially oriented then the solution at this region is wrong. Consequently the results at this area has to be rejected for post-processing tasks and analysis. Only the core of the domain is valid. That is what we call "measurement region".

The geometry generator permits periodic solid phase to be compatible with the periodic BCs. Both the periodic BCs and the free-slip impermeable BCs (mirror) have been compared. The domain size and the cell size used for the simulations comply with the restrictions forced by the previous studies. Figure

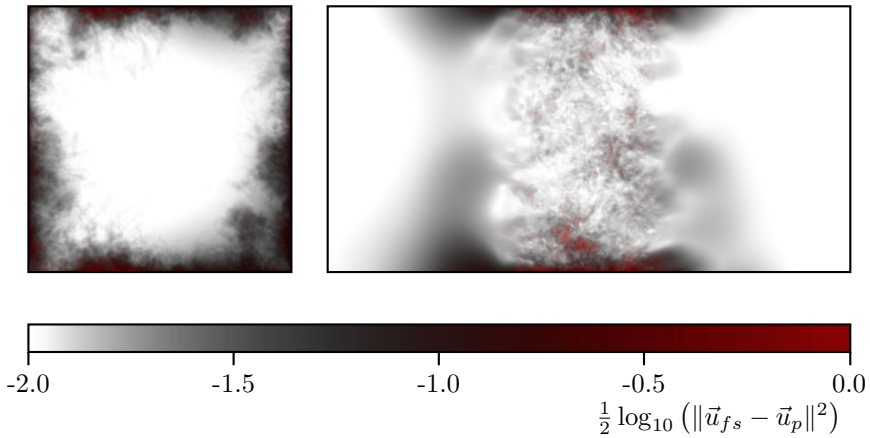

Figure 5.5: Volumetric render of the logarithm of the energetic norm of the error. (Left) Top view. (Right) Lateral view.

5.5 shows the module in log scale of the difference between the speed obtained from the simulation under the impermeable BC $\vec{u}_{fs}$ and the periodic one $\vec{u}_p$. White space indicates no difference, while red opaque colours indicate high differences. The black semitransparent regions show average differences. It is very easy to see how the impermeable BCs perturb the area at the perimeter. The strongest perturbations are at the boundaries and their effects decrease with the distance from the domain limits. Using an error threshold of 1%, it can be concluded that the solutions for both BCs are the same, around one correlation length from the wall. The effect can also be observed upstream and downstream from the porous medium. The lateral view of figure 5.5 shows it as smooth dark areas before and after the solid phase. The extension of the affected region is also around one correlation length up and down from the solid phase. As conclusion the closest threshold for the pressure BCs (upstream and downstream BCs) is located at one correlation length from the more prominent part of the solid phase.

The white space located far away from the solid phase means low difference in the net flux through the porous medium between both BCs. The domain is thick enough to dilute the perturbations produced by the presence of the unnatural limitations. Otherwise an uniform grey colour would appear instead. Because the affected area does not touch the centre and the net flux is almost the same for both simulations it can be concluded that the ducting effect

produced by the impermeable BCs has no consequences on the net flux.

The area distorted by the presence of the impermeable BCs should not be considered for local analysis purposes. The flux is forced to take an unnatural direction at this region. As a consequence, only the inner part is valid. We call this region the "measurement area". It is very interesting to see how its extension does not depend on the depth (Y direction) of the porous medium. It is constant with respect to this variable. It can be concluded that the limitation for the thickness of the domain is only related to the size of the structures of the porous media. Consequently the aspect ratio of the domain is not a parameter to take into account. The length in the pressure gradient direction and the thickness of the domain are independent variables. Only the size of the largest structure fixes their limits.

Measurement region

The measurement region is the inner region of the domain where the fluid pattern is not perturbed by the lateral BCs. It can be estimated by the simulations of a periodic domain with and without periodic BCs. Considering the periodic case as reference the difference between the two cases returns the error. At the limits the error has the same order of magnitude than the studied variable itself. It can be easily understood considering that the normal component to the limit does not exist for free-slip BCs. The error becomes smaller when the distance to the wall increases. The measurement area begins when the error is smaller than the threshold used as criteria. For the sake of simplicity we decided to use as measurement area the inner region of a parallelepiped concentric to the domain whose maximum error is lower than 1% times the reference velocity in every point.

5.2 Analysis of the results

There is no experimental setup which the results could be compare with. The scale of the problem is too small to capture any local detail in the flow field. Only macroscopic properties could be easily measured. Even so validation of macroscopic results requires the simulation of measured samples. Obtaining 3D micro-structures of porous material is very expensive and requires special installations like a synchrotron. Consequently the validation of the code has been realised in a different way. It has been studied the tendency of the relation between the pressure drop and the mass flow rate. If the simulations are correctly done the macroscopic results have to fit the theoretical results of Darcy equation.

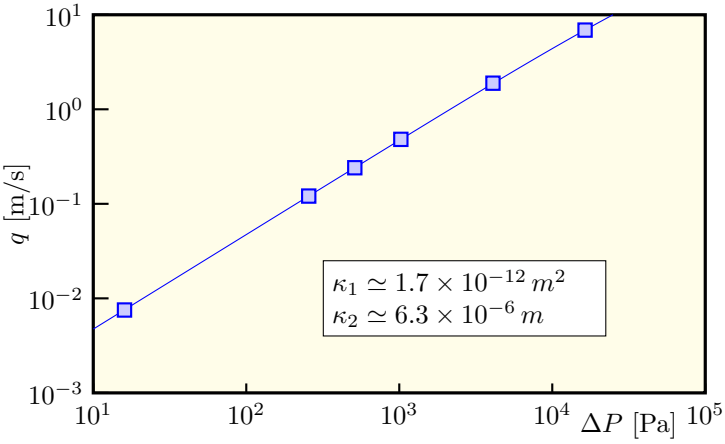

Figure 5.6: Pressure *vs.* velocity results for the same geometry. The fitting curve is represented by the continuous line. Magnitudes in logarithmic scale.

The simulation setup accomplishes the requirements imposed by the previous section. It is simulated a wall made of porous material. Lateral BCs are periodic as well as the micro-structure. The flow is driven by a pressure jump between the upper and lower limits of the domain. Several simulations are calculated for different pressure drops. Concretely the simulated values are 16, 256, 512, 1024, 4096 and 16384 Pa. It can be observed how the pressure jump follows a geometrical regression where some of the values have not been calculated. The covered range for this validation is larger following this procedure. Inertial effects are expected for the highest pressure. Forchheimer correction 2.2 considers the deviation from the linear behaviour due to this phenomenon.

The mass flow rate across the domain for every calculated pressure drop are showed in figure 5.6. With the results it is also plotted the fitting line obtained by least squares. It can be observed how the samples are practically aligned with the exception of the last one, when inertial effects take place.

Once the tendency has been validated it is time to analyse one of the cases with detail. Concretely the selected case is the simulation with 16kPa of pressure drop. No large difference has been observed with respect to the rest of cases with the exception of some fluid patters before and after the wall. The observations of this case can be extrapolated to the rest.

Figure 5.7 shows a cut plane of the solution for 16kPa. It can be observed in the results how the pressure gradient is not uniform across the porous media. The material can be described as a collection of pores connected by interstices.

Pores are like caverns in the porous medium. The flow is practically stopped there and the pressure is almost uniform in its interior. On the contrary the interstices, restrictions in the way of the fluid, concentrates the pressure drop and accelerates the flux.

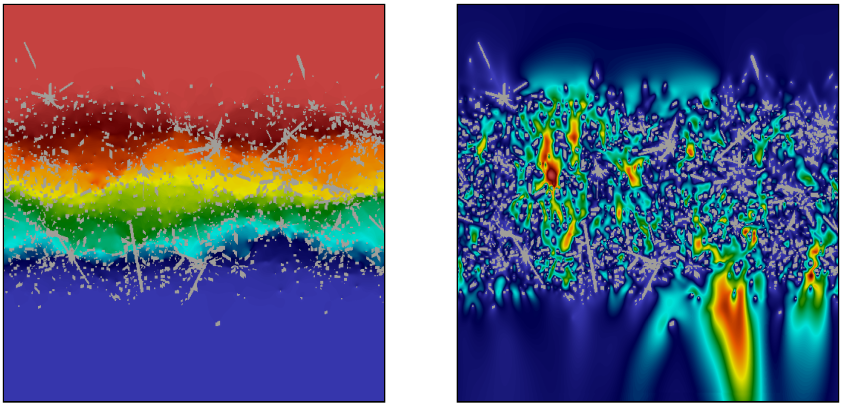

Figure 5.7: Results of an slide for 16kPa of pressure drop. (left) Pressure field. (right) Velocity magnitude field.

Bibliography

[1] Sukop, M.C.; Huang, H.; Lin, C.L.; Deo, M.D.; Oh, K.; Miller, J.D. Distribution of multiphase fluids in porous media: Comparison between lattice boltzmann modeling and micro-x-ray tomography. *Phys. Rev. E*, 77(2):(2008), 026710.

[2] Hayashi, H.; Kubo, S. Computer simulation study on filtration of soot particles in diesel particulate filter. *Computers & Mathematics with Applications*, 55(7):(2008), 1450 – 1460. Mesoscopic Methods in Engineering and Science.

[3] Bernsdorf, J.; Brenner, G.; Durst, F. Numerical analysis of the pressure drop in porous media flow with lattice boltzmann (bgk) automata. *Computer Physics Communications*, 129(1–3):(2000), 247–255.

[4] Rong, L.W.; Dong, K.J.; Yu, A.B. Lattice-boltzmann simulation of fluid flow through packed beds of spheres: Effect of particle size distribution. *Chemical Engineering Science*, 116:(2014), 508–523.

[5] Rong, L.W.; Zhou, Z.Y.; Yu, A.B. Lattice–boltzmann simulation of fluid flow through packed beds of uniform ellipsoids. *Powder Technology*, 285:(2015), 146–156. Pharmaceutical Particle Technology.

[6] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Lattice boltzmann simulation on porous structure and soot accumulation. *Mathematics and Computers in Simulation*, 72(2-6):(2006), 257–263.

[7] Yamamoto, K.; Satake, S.; Yamashita, H. Lattice boltzmann simulation on flow with soot accumulation in diesel particulate filter. *International Journal of Modern Physics C*, 18(4):(2007), 528–535.

[8] Yamamoto, K.; Satake, S.; Yamashita, H.; Takada, N.; Misawa, M. Fluid simulation and x-ray ct images for soot deposition in a diesel filter. *European Physical Journal-Special Topics*, 171:(2009), 205–212.

[9] Yamamoto, K.; Oohori, S.; Yamashita, H.; Daido, S. Simulation on soot deposition and combustion in diesel particulate filter. *Proceedings of the Combustion Institute*, 32:(2009), 1965–1972.

[10] Yamamoto, K.; Yamauchi, K.; Takada, N.; Misawa, M.; Furutani, H.; Shinozaki, O. Lattice boltzmann simulation on continuously regenerating diesel filter. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 369(1945):(2011), 2584–2591.

[11] Yamamoto, K.; Yamauchi, K. Numerical simulation of continuously regenerating diesel particulate filter. *Proceedings of the Combustion Institute*, 34:(2013), 3083–3090.

[12] Yamamoto, K.; Ohori, S. Simulations on flow and soot deposition in diesel particulate filters. *International Journal of Engine Research*, 14(4):(2013), 333–340.

[13] Stewart, M.; Maupin, G.; Gallant, T.; Zelenyuk, E.; Kim, D. *Fuel Efficient Diesel Particulate Filter (DPF) Modeling and Development.* Richland, Washington 99352: Pacific Northwest National Laboratory (2010).

[14] Falconer, K. *Fractal Geometry: Mathematical Foundations and Applications.* Wiley (2004).

[15] Block, A.; Vonbloh, W.; Schellnhuber, H. Efficient box-counting determination of generated fractal dimensions. *PHYSICAL REVIEW A*, 42(4):(1990), 1869–1874.

Chapter 6

Conclusions and future works

6.1 Conclusions

The thesis is focus on the analysis of the filtering capabilities at micro-scale level of DPF. Several models and solvers have been successfully developed to reproduce the required multi-physics. Each physical phenomena have been developed independently and included in a module. This strategy facilitates the validation of each part without any influence of the rest.

The developing process has been set out as an optimisation problem where the objective to be optimised is to advance as much as possible. The reason for this approach is the complexity of the simulations. It is unreachable for a single thesis. This kind of simulations includes fluid dynamics of the air through the filter, particle motion, particle deposition and regeneration. Regeneration was ruled out from the beginning. With respect to the rest of the models, deposition could not be developed on time. Fluid dynamics and particle motion have been successfully validated.

Every optimisation process is subjected to restrictions. In this context the restrictions are the available resources, which are limited, and the quality of the results. Simulating a filter is computationally expensive. It has to be realised in a workstation, where the number of cores and the available memory are limited. For this specific application memory consumption is specially important. It is the most limiting factor. The algorithms have been adapted in consequence. Their design is a compromise solution between calculation

speed and memory consumption.

The research realised for this thesis contributes to science in two different ways. One of them is mathematics. The algorithms optimises the calculation and gives maximum performance. The other is knowledge related with the filtering process. Setup for porous media calculation with minimum computational cost, correlations and models of agglomerates or flow field in the interior of a chaotic micro-structure are three examples.

6.1.1 Geometry generator

The first task to do when it is wanted to measured the properties of a porous material is to obtain a sample of it. The micro-structure can be obtained by tomography or imaging reconstructions when the material and the facilities are available. The cost of these techniques is usually very expensive. The alternative is to create the geometry computationally. A tool for this purposed has been developed successfully. It is based on procedural generation techniques. The result mimics the micro-strucuture of a filter made of acicular mullite. Each crystal is characterised by a set of parameters. These parameters are random samples extracted from statistical populations.

6.1.2 Fluid Solver: Lattice-Boltzmann Method

Once the geometry is available it is time to calculate the flow trough it. Several solvers based on lattice Boltzmann techniques have been developed. One of these solvers is the test environment. It has been developed to be run in the GPU. It returns maximum performance. The simulations are calculated in minutes. The main purpose of this application is to test any advance before being included in the main solver. The Lattice Boltzmann solver LaBMoTer is the most advance tool developed in this thesis. This application includes adaptive refinement for both the geometry and the solution. It has been designed to be run in the workstation. The source code has been written in Fortran for maximum performance. HPC techniques have also been applied for the same purpose. LaBMoTer has been successfully validated with two reference cases, the 3D driven cavity and an 3D infinite circular cylinder at Reynolds 250. Part of the research has been realised in collaboration with the department Lehrstuhl für Systemsimulation (LSS) of Friedrich-Alexander Universität Erlangen-Nürnberg (FAU). This department, directed by professor Ulrich Rüde, develops waLBerla framework. This is the third tool whose purpose is calculating the flow field. waLBerla is designed to be run in the most powerful supercomputers. Its objective is to obtain maximum performance. It is widely validated and its range of applicability is very extensive.

This thesis contributes to the development of Lattice-Boltzmann Methods in two different ways. Boundary conditions in LBM are usually underrestricted. There are more unknowns than restrictions. Consequently there are infinite combinations of discrete distributions able to satisfy each boundary condition. The opposite case could also happens if the number of components in contact with the boundary is very low. A new technique to develop boundary conditions for LBM has been successfully tested. Each boundary condition is considered as an optimisation problem. The magnitude to be optimised depends on the criterion of the developer. Performance, accuracy and robustness are the most common. The type of boundary conditions imposes the restrictions, usually density (pressure) or velocity, or a combination of them. Lagrange multipliers are applied for the optimisation problem, resulting an equation system. The uniqueness and existence of the solution are not guaranteed. A robust algorithm able to work with over-determine systems is required. It returns the best approximation to the solution with minimum euclidean norm, exact solution if the system is determined. This algorithm is computationally expensive (the computational cost is similar to other BCs if the inverse of the equation system is kept in memory) but it is robust for every case. Its main advantage in addition to its robustness is the absence of conditional flow controls.

The experience of developing applications based on LBM with different strategies also provides important information about performance. Maximising performance without loss of accuracy was a design criterion from the beginning. This is the reason why LaBMoTer incorporates local adaptive refinement for both the solution and the boundary conditions. However after its development it was discovered how the interpolations and the communication between different lattices deteriorate the performance. In addition local refinement breaks the coherent arrangement of the data in memory. It slows down the calculation speed even more. waLBerla framework solves this issue using block refinement instead of local refinement. The number of nodes for each sub-block is always the same, keeping good load balance in parallel computation. The negative counterpart of this strategy is an increment in the number of cells. However the advantages outweigh the disadvantages. A deeper improvement is reachable by vectorising the collision step. Vectorised operators are able to operate with arrays instead of scalars with the same frequency. Consequently it is possible to calculate and array of nodes in the same time than a single one. It is recommendable consistent memory alignment to obtain maximum performance. Calculations in the GPU are specially sensible to this paradigm, where the improvement could be of one order of magnitude or even more. This is in fact how the test environment works.

6.1.3 Optimum setup

Obtaining good results is only possible with a good setup. Domain size, discretisation and boundary conditions are the most conditioning factors. To minimise the computational cost the domain size has to be minimised, the lattice spacing maximised and the boundary conditions have to be as realistic as possible. It has been simulated the pressure drop of an infinite wall made of porous material. The domain takes the shape of a box. Two of the boundary conditions force the flux through the porous media. The rest (lateral BCs) communicates the simulated sample with the rest of the material. There are several ways to delimit the sample. The most appropriate one is periodicity. This boundary conditions requires periodic geometry. These is an advantage of the procedural-generated micro-structures in comparison with measured micro-structures. Forcing periodicity in a generated geometry is something very easy to do. On the contrary is very difficult to find pure periodic micro-structures in nature. If it is not possible to use periodic boundary conditions the second option is symmetry. Symmetry artificially channelises the flow. It is important to known how deep penetrates the influence of the lateral boundary conditions to take it into consideration with non-periodic geometries. The influenced region can never touch the core. It has been demonstrated how the influence distance is proportional to the correlated distance. Autocorrelation has also been used to calculate the size of the largest coherent structures. The correlated distance duplicates the size of the structures. The domain has to contain several of this structures to be statistically representative. The last factor to be analysed in the setup is lattice spacing. To reduce the computational effort the lattice spacing has to be maximised preserving the accuracy. Two procedures have been used to calculate the appropriate lattice spacing. The fractal dimension of the interface takes value two (a surface) when the resolution is good enough to capture the smallest details. It has been used the box-counting algorithm for this purpose. The result has been corroborated with a lattice-size independent study. Multiple resolutions have been applied to solve the same geometry, a reduced version of the definitive one. The resolution is considered appropriate when finer lattices have no influence in the solution.

6.1.4 Fluid field solution

The previous described setup has been used for the final simulation. The results provide very interesting information about the flow field and the micro-structure. It has been discovered how the flow field in the interior of the filter can be approximated by pores connected by interstices. It can be seen by the patterns of the pressure and velocity fields. The pressure drops in the interstices whereas it is uniform in the pores. It has been visualised with a volumetric render whose opacity is proportional to the pressure gradient. The

results of the velocity field show a similar pattern. The highest velocities are in the interstices. Unlike the pressure field in the interior of a pore, the velocity field is not uniform. High velocity regions connect the interstices creating vessel patterns, specially for high speeds. It has also been discovered how most of the mass flow rate crosses through the same interstice. It can be visualised tracing the path-lines. It means that the mass flux is dominated by the largest interstices whereas the lowest have not practically effect.

6.1.5 Particle Motion

Once the flow field is successfully solved the particles can be injected in the domain. The size of Diesel particles are not directly measured. They are characterised by the equivalent aerodynamic diameter. This magnitude explains the aerodynamics of the particles perfectly. However it is not applicable when other effects have to be considered. This is the case of the smallest and more numerous particles. Brownian motion dominates their trajectory. The data are in equivalent aerodynamic units. However Brownian motion requires physical units. It is necessary to find the equivalence. For such purpose two different solvers have been successfully developed. Diesel particles can be approximated by aggregates of spheres. The first solver simulates the aggregation process returning populations of particles. The second solver calculates the aerodynamics of the generated particles. Unfortunately it is not finished.

Ballistic Agglomeration Simulator

The simulator of the aggregation process is based on ballistic collisions. Advance divide and conquer algorithms have been developed *ad hoc* to accelerate the calculation. In addition, when there is an impact between agglomerates, they can fuse to create a bigger one or they can collide. Both cases follows the Newton laws. Usually the formulation to calculate the post-collision velocities are based on the restitution coefficient. It explains the difference in speed before and after collide. This option is the most typical. However it presents numerical problems due to round-off errors for very small denominators. It introduces extra energy in the system and makes the motion unnatural. A novel formulation developed in this thesis has been successfully implemented. Instead of using the restitution coefficient and calculate directly the post-collision the new technique is based on energy. Every impact produces an impulse. This impulse is unknown at the beginning. On the contrary the post-collision energy can be easily calculated and written in terms of the unknown impulse. Consequently the impulse can be obtained solving a second order equation and choosing the appropriate solution. Once the impulse is known the linear and angular velocity can be refreshed. The energy has a second purpose. It has

been used to calculate the discriminant responsible to decided when the colliding agglomerates fuse and when they collide. It is expressed as a probabilistic function. The probability to collide or fuse depends on the energy absorption required for fusion. Once the probability is calculated a random sampled is extracted from a uniform distribution between 0 and 1. If the value is larger than the probability the agglomerates collide, otherwise they fuse.

Analysis of Generated Particles

The solver has been used for two different purposes. The first and design reason is to obtain a representative population of particles. The second applicability is generation of porous media created by depositions. The first set of simulations has been done starting from a population of spherical particles in full-periodic domain. The size of the precursors is randomly generated accordingly to a statistical population. Their velocity is initialised in a similar way. The simulation finishes when the number of particles is less or equal than 128. Several magnitudes of the results are evaluated. The agglomerates are self-similar for large scales. Their compactness can be quantified by means of the fractal dimension D_f. This magnitude has been calculated returning the value 1.9. It relates the number of precursors n_p of an agglomerate with its gyration R_g radius as $n_p = k_p \left(R_g/R_0\right)^{D_f}$. The pre-multiplicative factor k_p presents a lot of variability. This uncertainty has been quantified and include in the model. In addition the evolution of concentration of particles with time has been also studied. A decay equation has been modified to include the effects of agglomerate growing. It depends on the initial concentration of precursors, restitution coefficient, variation of precursor radii, energy absorption for equal probability between fusion and collision and overlapping energy absorption range for coexistence of fusion and collision. Random combinations of these parameters have been created by means of Lattice-Hypercube techniques. Each combination is used as input for the corresponding simulation. The evolution in time in the number of particles is saved to fit the modified decay equation. Each combination is associated with the resulting fitting parameters. After simulate every case the results can be used to find a model able to explain the fitting parameters by means of the inputs. However the agglomeration process presents natural variability. It has a chaotic component. The effects of chaos can be seen in the residuals, too high to be neglected. The model has to include uncertainty. Four stochastic variables are added to the model. As consequence every combination of inputs returns a probabilistic distributions of fitting parameters.

Agglomerate Shell Generation

The agglomerates are modelled by spheres. However to calculate the aerodynamics it is needed to know where is the surface. A novel algorithm has been developed for this purpose. It is based on meta-spheres. This strategy consists on replacing every sphere by a potential. The original geometry can be recovered as an iso-surface of this potential. This technique rounds the intersections between spheres, making the geometry more realistic. Several points are located on the iso-surface. The separation among the points is uniform. The surface can be constructed using Delaunay triangulation. Some more improvement is required to be fully operative.

Model for deposition structure

The agglomerate solver based on ballistic collision can be also applied to create porous media resulting of depositions. The simulation is realised in a very long box-shaped domains. The domain is periodic along the two shortest directions. The third direction is connected to a reservoir of precursors in one of its sides (inlet) and to an empty space (outlet) in the other. When an agglomerate collides with its twin of the neighbour period it becomes solid phase. Unfortunately the resultant geometry can not be used to characterise deposits of particles for Diesel filters. However it can be considered as the starting point of the definitive solver. Realistic solutions are only possible if it is also considered fracture models for agglomerates and deposits.

Aerodynamics of agglomerates: Direct Simulation Monte Carlo

Now it is possible to relate the number of precursors (particle mass) with the gyration radius (particle size). The following step is to find the relation between the number of precursors and the equivalent aerodynamic diameter. This model allows the capability to convert the measurements of the bibliography, given in equivalent units, to physical units. Relating the aerodynamic properties and the real geometry is only possible by means of simulations. For such purpose a new solver has been developed. Conventional CFD techniques like Finite Volumes or the solver developed to compute the flow field of the porous media based on lattice Boltzmann Methods are accurate for continuous flows. However the size of the precursors is comparable to the mean free path. Consequently a new solver able to calculate rarefied flows is required. The most appropriate option is an application based on DSMC. The solver has been successfully developed, including a new wall boundary condition. The boundary condition is regulated by two parameters. The calibration has been done simulating laminar Couette flow. For validating purposes the drag of a sphere has

been successfully calculated. Unfortunately the wall boundary condition for triangulated surfaces could not be implemented on time. Consequently finding the relation between the geometry of the agglomerates and the equivalent aerodynamic diameter is one of the most important future works. The process to be realised is analogous to the method followed to find the relation between the gyration radius and the number of precursors of an agglomerate.

Particle Tracker

The algorithms to calculate the trajectory of the particles have been successfully developed. The starting point is the model to calculate the path-lines. Path-lines and trajectories overlap if the flow field does not depend on time and if the inertial effects are negligible in comparison with the drag force. The range of applicability of this model is reduced. Including the mass extends the applicability interval. The middle range of the particle population can be successfully tracked with this model. However the smallest particles are also affected by Brownian motion. Brownian motion is the discrete counterpart of the Fick's law for diffusion. This equivalence has been used to validate the model. Sometimes the displacement due to Brownian motion is too long. If the covered distance during a time step is longer than the cell size the Brownian motion is decomposed in several sub-steps. With respect to the largest particles, they can be several cells thick. The drag force and its corresponding reaction force are calculated by a kernel function. In addition, the reaction force could be high enough to disturb the flow field around them. The stationary hypothesis is not valid under this condition. Fluid field and particle tracking are couple.

6.2 Future works

Perfection does not exist. Every developed model, solver and algorithm of this thesis can be improved. However there are some points where inverting effort for improvement could be more effective. In addition some parts of the thesis are unfinished. In the following lines it is explained how some of the most relevant research can be concluded, improved, adapted or expanded. The explanation summarises the knowledge provided by four years of experience.

6.2.1 Solvers, models and algorithms

An algorithm has to be effective and efficient. Efficiency is usually measured in terms of performance (maximum calculation speed or scalability) or consump-

tion of resources (minimum required memory or energy consumption). The definition is usually related with the available resources or with a magnitude to be optimised. Effectiveness is defined by the objective of the algorithm, the reason of its creation. It measures the quality of the results. An algorithm is effective if it accomplishes its purpose. The most typical variables to quantify effectiveness are accuracy (simulation), robustness or limited computational time (real time applications). An improvement in one of these factors usually penalises others. The election of the best options is then a compromise solution. In the following lines it is exposed some alternatives to the algorithms used in this thesis and their potential improvements.

Lattice Boltzmann Solver

The first version of the solver LaBMoTer has been developed to be run in a shared memory machine. One of its more important capabilities is adaptive refinement. The lattice around the geometry or in the high gradient regions can be made finer to capture the details. This strategy saves memory without loss in accuracy. The refinement of the lattice is done locally. This has a penalty in performance. Firstly the communication between coarse and thin lattices requires interpolation, which is a very expensive operation. The contact area between lattices can vary with time and be very extensive.

One advantage of the modern processors is the capability to overlap calculations and memory transfer, what is called prefetching. It reduces drastically the cache misses and consequently it has beneficial effects in performance. There is a requirement for prefetching to be effective. The location in memory of the data has to be known *a priori*. Local refinement breaks the coherent arrangement of the data. Knowing their location is more difficult and prefetching becomes less effective. The addition of all the previous effects slows down the solver. Even so using refinement is advantageous in comparison with full fine lattice.

The domain in LaBMoTer is stored in memory as a single block. To be able to run in distribute memory machines the domain has to be divided in multiple blocks. This is in fact how waLBerla framework works. Each block is calculated by a single computational thread. To get maximum performance each blocks contains a single lattice. In other words, refinement is done for entire blocks instead to be local. This is done in a recursive way creating octree structures, where each children contains the same number of nodes than the father. Preserving the number of nodes keeps good load balance. It makes the algorithm highly parallelisable. In addition the data in each block is arranged. The access is fast and prefetching applicable. Performance increases in one order of magnitude.

Additional improvements in the performance can be obtained vetorising the arithmetic operations. This is the strategy followed to developed the test environment. This technique is specially useful to calculate with the GPU, when the performance improvement is more than an order of magnitude. The improvement for the CPU is not so drastic. However it can doubles the calculation speed. The modern CPUs are able to apply the same arithmetic operation on data arrays simultaneously. How long is the array depends on the CPU architecture. This type of operations are called SIMD, an acronym Single Instruction, Multiple Data. The inverted time to calculate the entire array is approximately the time required to calculate a single scalar.

It is proposed to developed a new solver LaBMoTer 1.0 which includes these strategies to improve drastically the performance.

Ballistic agglomeration simulator

The size of the particles is heterogeneous. The range of precursors per agglomerate goes from a single one to ten of thousands. Calculating collision between agglomerates requires to know where is the exact location of the impact. The algorithm which detects the collision points is divided in two different stages. The first stage calculates proximity. Once proximity is confirm the second stage calculates the contact point. The agglomerates are circumscribed by spheres. Two agglomerates are considered candidates to collide if the circumscribed spheres intersect each other. Every combination of agglomerates has to be tested. The cost of the computation is quadratic. Two different strategies are used depending on the number of agglomerates in the domain. The brute force algorithm is the best option for low number of particles. The cost grows quadratically. When the number of agglomerates excesses a threshold the algorithm to be applied follows a divide and conquer strategy. The computational cost is reduced from quadratic to $\mathcal{O}(N \log(N))$. The domain is recursively divided creating an octree structure. This decomposition is created every iteration.

The method to find the location point once a couple of agglomerates are candidates to collide is similar. The algorithm to be applied when the agglomerates are small is the brute force version. Again, the order of the second phase is quadratic. A second divide an conquer algorithm divides the agglomerates to find the location of the collision point. As well as the other divide and conquer algorithm the order is reduced to $\mathcal{O}(N \log(N))$. Again the decomposition is done for every iteration.

It is propose as a future work to create and maintain the octree mesh. The current algorithms are designed to run in share memory machines. Keeping the mesh allows the distribution of the computational load among different nodes.

In addition both the proximity detection and collision detection can be unified. It can drastically reduce the heterogeneity and accelerate the calculation.

Triangulation of the surface of the agglomerates

Obtaining the shell which surrounds the agglomerates have been partially developed. The procedure consists on using meta-spheres. A potential field created by kernels is used to generate the shell which encloses the agglomerate. An iso-surface of this potential corresponds with the surface of the agglomerate. The location of the vertexes along the surface is uniform and it is correctly computed. Once the vertex are located it is time to create the triangles. Delaunay triangulation is the selected tool for this purpose. Unfortunately some of the generated triangles are not part of the geometry. The opposite could also happens. The shell could present some holes where a triangle should exist. Some algorithms have been developed to correct the false positives and to close the holes. However they only correct the geometry partially. More advance algorithms are required. It is proposed as a future work to develop these algorithms.

Improvement of the micro-structure deposits

The current model creates micro-structures with self-similarity relations. Its fractal dimension is expected to be coincident with the fractal dimension of the agglomerates generated by ballistic collision. It means that large micro-structures have proportional empty spaces. In the real world these micro-structures collapse. The links between precursors can break. Modelling this effect reduces the size of the empty spaces. It is purposed as future work to developed a model of rupture when two agglomerates collide.

Direct Simulation Monte-Carlo

The development of the CFD solver based on DSMC is not finished. This is the tool designed to calculate the aerodynamics of the agglomerates. The formulation has been successfully validated. However it is still required to develop the boundary conditions for triangulated surfaces.

The solver can be improved and its capabilities extended. More advance collision operators can be implemented to gain in performance. In addition the arithmetic can be optimised from the computational point of view. At this moment the solver does not include adaptive refinement. It is propose as a future work to improve the algorithms of the solver and extend its capabilities

including adaptive refinement. Developing the boundary condition for triangulate surfaces is mandatory. Otherwise it is not possible to calculate the drag force of the agglomerates.

The agglomerates are not symmetrical. The orientation is aleatory. The aerodynamic forces tend to orientate the agglomerate to minimised the drag. Implementing moving walls could help to calculate minimal drag force. The agglomerate has to be free to rotate and been oriented with the wind. It is proposed to implement moving walls to calculate minimum drag instead of the drag aligned with the initial orientation.

6.2.2 Extra simulations

Deposits characterisation

The ballistic agglomeration simulator can be used to create the micro-structure of deposits. To obtain realistic results it is needed to previously develop the rupture model (see section 6.2.1). Once the micros-structure is created and its surface triangulated the flow field trough the deposits can be simulated. The scale of the details is similar to the length of the mean free path, so the simulation has to be realised with the DSMC solver. The results can be used to characterised the material. The macroscopic properties such as porosity and permeability can be obtained following the same procedure of this thesis to analyse acicular mullite. Then the deposits can be integrated in the filter simulation as a porous phase.

Deposits as porous phase

Once the deposits are well characterised they can be included in the filter simulation as a porous phase. The Darcy equation has to be converted to Lattice Boltzmann formulation just like it is done with NS equation. It is also required to model the growing of the deposits and the behaviour of the interface fluid-deposit.

Aerodynamic characterisation of particles

Direct-Simulation Monte-Carlo can be used to calculate the aerodynamics of the particles. The number of simulations has to be large enough to be representative. Correlations between the drag force and the real size of the particles can be use to create a model in analogous way to the work done in section 4.3.4.

This model relates the the physical size, required for Brownian motion, with the equivalent aerodynamic diameter. The measured populations are usually given in equivalent units. This tool can be used to convert the measurements.

Simulations of other micro-structures

This thesis is focused on the simulation of acicular mullite. There are Diesel Filters made of other materials. The topology of these materials can be very different. It is proposed as future work to simulate other micro-structures following the same procedure. In addition other procedural-generation algorithms can be developed to mimic the corresponding materials. The study of the optimum setup can be used to corroborate and generalise the results obtained in this thesis.

Simulations with oriented crystals

All the simulations of this thesis calculate isotropic material. Under this hypothesis the permeability tensor can be replaced by a scalar. It could be interesting to recover the macroscopic properties of anisotropic materials. It is propose to modify the crystal generator to create oriented structures and its posterior analysis. Several simulations are required instead of a single one to analyse every direction. It is also necessary to calculate the auto-correlation for the third direction too. The best option for the analysis of this type of materials is calculating a 3D sample instead of a fragment of wall. Consequently is advantageous to create a full-periodic geometry and to developed a new boundary conditions. This new boundary condition has to satisfy periodicity in speed and force net mass flow rate or pressure gradient.

Geometry depending on depth

All the simulations of this thesis analyse pure chaotic micro-structure of acicular mullite. The properties of the material are statistically uniform. It is propose to modify the geometry generator to create depth-dependent material. The most appropriate magnitude to be modified is crystal length. The results can be compared with the results of uniform materials and find some equivalence.

Relation between macroscopic properties and micro-structure

In this thesis a new model has been developed to relate the number of precursors of an agglomerate with its gyration radius. Similarly a second model is able to predict the evolution of the concentration of agglomerates with respect to time. A third model is proposed as a future work to relate the physical size of the particles with its aerodynamic properties. The same methodology can be used to predict the macroscopic properties of a material from the parameters which control its micro-structure. The auto-correlation and cross-correlation could be also variables to take into account for both correlating and being correlated. Several simulations of variable versions of the geometry are proposed as future work to find this relation between.

Appendix A

List of Symbols

A.1 Acronyms

List of acronyms

BC	Boundary Condition
BGK	Bhatnagar-Gross-Krook
CDF	Cumulative Distribution Function
CFD	Computational Fluid Dynamics
CMT-Motores Térmicos	Centro de Mantenimiento de Transporte
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DDR	Double Data Rate
DNS	Direct Numerical Simulation
DPF	Diesel Particulate Filter
DSMC	Direct Simulation Monte-Carlo
DoF	Degrees of Freedom
FAU	Friedrich-Alexander Universität Erlangen-Nürnberg
FFT	Fast Fourier Transforms
FVM	Finite Volume Method
GPGPU	General-Purpose computing on Graphics Processing Units
GPU	Graphics Processor Unit
HPC	High Performance Computing
IFFT	Inverse Fast Fourier Transforms
ITWM	Fraunhofer Institute for Industrial Mathematics
LBM	Lattice Boltzmann Method

LES	Large Eddy Simulation
LSS	Lehrstuhl für Systemsimulation
MD	Molecular Dynamics
MPI	Message Passing Interface
MRT	Multiple Relaxation Time
NPNL	Northwest Pacific National Laboratory
NS	Navier-Stokes
OpenMP	Open Multi-Processing
PAH	Poly-cyclic Aromatic Hydrocarbons
PDF	Probability Density Function
PM	Particle Matter
RAM	Random Access Memory
SIMD	Single Instruction, Multiple Data
SRT	Single Relaxation Time
TRT	Two Relaxation Time
UQ	Uncertainty Quantification

End of list of acronyms

A.2 Mathematical symbols

The square $\square$ replaces a generic variable.

List of symbols

Roman symbols

k_B	Boltzmann constant
$\mathscr{D}_{MB}$	Minkowski-Bouligand fractal dimension
D_f	Fractal dimension
D	Diameter
$\mathscr{D}$	Diffusion coefficient
E	Energy
$\vec{F}$	Force
F_D	Drag Force
$[I]$	Unitary Matrix
Kn	Knudsen number
$\vec{L}$	Angular momentum
P	Pressure
Re	Reynolds number
St	Strouhal number
T	Temperature

a, b, c	Polynomial coefficients
$\vec{a}$	Acceleration
A	Area
$\vec{c}$	Microscopic velocity
C_D	Drag coefficient
C_R	Restitution coefficient
C_C	Correction factor
D	Number of dimensions
$\vec{d}$	Distance
f	Botzmann distribution
$\vec{F}_b$	Buoyancy force
$\vec{F}_B$	Brownian force
$\vec{F}_g$	Gravity force
f	Frequency
$\mathscr{F}$	Function to be minimised
$\mathscr{G}$	Restriction function
g	Gravity constant
$G(\square;\square)$	Green's function
h	Lattice spacing
k_f	Pre-multiplicative constant
K	Refinement factor
K	Kernel function
L	Characteristic length
$\mathscr{L}$	Linking function
m	Discrete independent variable
m	Mass
n	Discrete independent variable
$\vec{n}$	Normal vector
N_{Br}	number of sub-time steps for Brownian motion
n_p	Number of precursors
$\mathbb{N}$	Natural numbers
$\vec{p}$	Linear momentum
q	Dynamic pressure
q	Mass Flux
$\mathscr{R}_f$	Autocorrelation of f
$\vec{r}_i$	Eigenvector
r	Radius
R_g	Gyration radius
$\mathbb{R}$	Real number
$[R]$	Rotation matrix
s	Scale factor
S	Reference Area
t	Time
$\vec{u}$	Macroscopic Velocity
U	Characteristic velocity

w	weight
$\vec{x}_{cm}$	Centre of Mass
z	Normalised independent statistical variable

Greek symbols

Π	Momentum flux tensor
Ω	Collision operator
χ	Random sample
$\delta_{\square\square}$	Kronecker's delta
ε	Infinitesimal
ε	Error
κ	Permeability
κ_1	Inertial permeability
$\kappa_\square$	Runge-Kutta intermediate step
κ	Wave number
λ	Lagrangian multiplier
λ	Eigenvalue
λ	Mean free path
μ	Dynamic Viscosity
ϕ	Porosity
ϕ	Phase field
π	Quotient between perimeter and diameter of the circle
ρ	Density
σ	Standard deviation
τ	Relaxation time
θ	Temperature
τ_d	Characteristic time
ξ	Spatial displacement
$\vec{\omega}$	Angular velocity

Operators

$\boldsymbol{f}$	Array (bold variable)
$\lceil\square\rceil$	Upper integer
$\square^*$	Complex conjugate
$\square \star \square$	Crosscorrelation
$\square'$	Deviation
$\vec{\nabla} \wedge \square$	Curl
$\Delta\square$	Increment
$\vec{\nabla} \cdot \square$	Divergence of
$\vec{\nabla} \otimes \square$	Gradient of
$\partial_x\square$	Partial derivative with respect to x
$\partial\square/\partial x$	Partial derivative with respect to x
$\dot{\square}$	Time derivative
$\ddot{\square}$	Second time derivative

$\mathrm{erf}(\square)$	Error function
$\lfloor \square \rfloor$	Lower integer
$\hat{\square}$	Fourier transform
∞	Infinite
$\int \square \, dx$	Integral with respect to x
$\|\vec{\square}\|$	Magnitude
$[\square]$	Matrix
$[\![\square]\!]$	Tensor
$\bar{\square}$	Normalised variable
$\square_{\bar{\square}}$	Component of opposite direction
$\mathcal{O}(\square)$	Order of
$\vec{\square} \wedge \vec{\square}$	Cross product
$\vec{\square} \cdot \vec{\square}$	Dot product
$\vec{\square} : \vec{\square}$	Internal tensor product
$\vec{\square} \otimes \vec{\square}$	External tensor product
$\sqrt{\square}$	Root square
$\square^{\otimes \square}$	Tensor power
$\vec{\square}$	Vector

Subscripts

$\square_0$	Reference value
$\square_\alpha$	Relative to the α-th component of the distribution
$\square_c$	Relative to the coarse level
$\square_{D\&C}$	Relative to divide and conquer algorithm
$\square_f$	Relative to the fine level
$\square_k$	Restriction Index
$\square_p$	Relative to the particle
$\square_r$	Relative magnitude
$\square_t$	Relative to time
$\square_w$	Relative to the wall

Superscripts

$\square^{\mathrm{eq}}$	Relative to equilibrium
$\square^{\mathrm{neq}}$	Relative to non-equilibrium
$\square^{\mathrm{eq}}$	Equivalent
$\square^{\mathrm{in}}$	Relative to incoming variables
$\square^{\mathrm{kn}}$	Known value
$\square^{\mathrm{out}}$	Objective value
$\square^{\mathrm{Ph}}$	Magnitude in physical units
$\square^{\mathrm{ref}}$	Reference value

End of list of symbols

Dictionary

A

azimuth, angle of: angle formed between the line of sight projected into the horizontal plane and the north.

E

elevation, angle of: angle formed between the line of sight and the horizontal plane.

Equivalent aerodynamic diameter: Given a generic-shaped particle, the equivalent aerodynamic diameter is the hypothetical diameter of a spherical particle with the same motion in Stokes regime with density 1000kg/m^3.

Equivalent particle: Given a generic-shaped particle, its equivalent counterpart is a spherical particle of density 1000kg/m^3 whose motion in Stokes regime is the same than the original one.

N

nadir: point of minimum elevation located vertically below the observer.

P

prefetching: Computational technique which consists on overlapping data transfer and calculations to reduce computational time.

V

voxel: is the homogeneous in density non-divisible unit of volume which compounds a 3D image.

Z

zenith: point of maximum elevation located vertically above the observer.

End of list of dictionary