VITOR MACIEL VILELA FERREIRA

A hybrid LES / Lagrangian FDF method on adaptive, block-structured mesh

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2015

VITOR MACIEL VILELA FERREIRA

A hybrid LES / Lagrangian FDF method on adaptive, block-structured mesh

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA**.
Área de concentração: Transferência de Calor e Mecânica dos Fluidos.

Orientador: Prof. Dr. Aristeu da Silveira Neto
Coorientador: Prof. Dr. João Marcelo Vedovoto

Uberlândia - MG

2015

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

F383h
2015

Ferreira, Vitor Maciel Vilela, 1986-
A hybrid les / lagrangian fdf method on adaptive, block-structured mesh / Vitor Maciel Vilela Ferreira. - 2015.
107 f. : il.

Orientador: Aristeu da Silveira Neto.
Coorientador: João Marcelo Vedovoto.
Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Mecânica.
Inclui bibliografia.

1. Engenharia mecânica - Teses. 2. Fluidodinâmica computacional - Teses. 3. Escoamento multifásico - Teses. 4. Monte Carlo, Método de - Teses. I. Silveira Neto, Aristeu da, 1955- II. Vedovoto, João Marcelo. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

CDU: 621

To my mother, Carla;

and my grandmother, Ignez.

ACKNOWLEDGEMENTS

Sou grato a Deus por sempre me fascinar com os detalhes da criação e por me instigar a conhecê-los. *“Through You I can do anything, I can do all things; cause it’s You who give me strength, nothing is impossible”*.

Agradeço à Faculdade de Engenharia Mecânica – UFU, que através do Programa de Pós-graduação me recebeu como mestrando; ao CNPq, a CAPES, a FAPEMIG e a Petrobras pelo apoio financeiro; ao professor Aristeu pelo ensino e confiança; ao professor João Marcelo pela orientação e revisão do trabalho; à Millena, representando todos aqueles que têm desenvolvido o código AMR3D, pelo apoio; aos colegas e amigos do MFLAB e da FEMEC.

Minha gratidão à família por viver comigo as alegrias e angústias deste tempo; em especial a minha esposa, Mariana, por compartilhar deste sonho e se comprometer com ele. *“Die empty”*.

"Long ago, even before he made the world God chose us to be his very own, through what Christ would do for us; he decided then to make us holy in his eyes, without a single fault — we who stand before him covered with his love. His unchanging plan has always been to adopt us into his own family by sending Jesus Christ to die for us. And he did this because he wanted to!"

Ephesians 1.4,5

FERREIRA, V. M. V., **A hybrid LES / Lagrangian FDF method on adaptive, block-structured mesh.** 2015. Master Thesis, Universidade Federal de Uberlândia, Uberlândia.

ABSTRACT

This master thesis is part of a wide research project, which aims at developing a computational fluid dynamics (CFD) framework able to simulate the physics of multiple-species mixing flows, with chemical reaction and combustion, using a hybrid Large Eddy Simulation (LES) / Lagrangian Filtered Density Function (FDF) method on adaptive, block-structured mesh. Since mixing flows provide phenomena that may be correlated with combustion in turbulent flows, we expose an overview of mixing phenomenology and simulated enclosed, initially segregated two-species mixing flows, at laminar and turbulent states, using the in-house built AMR3D and the developed Lagrangian composition FDF codes. The first step towards this objective consisted of building a computational model of notional particles transport on distributed processing environment. We achieved it constructing a parallel Lagrangian map, which can hold different types of Lagrangian elements, including notional particles, particulates, sensors and computational nodes intrinsic to Immersed Boundary and Front Tracking methods. The map connects Lagrangian information with the Eulerian framework of the AMR3D code, in which transport equations are solved. The Lagrangian composition FDF method performs algebraic calculations over an ensemble of notional particles and provides composition fields statistically equivalent to those obtained by Finite Differences numerical solution of partially differential equations (PDE); we applied the Monte Carlo technique to solve a derived system of stochastic differential equations (SDE). The results agreed with the benchmarks, which are simulations based on Finite Differences framework to solve a filtered composition transport equation.

Keywords: Mixing flow, Large Eddy Simulation, Filtered Density Function, Monte Carlo method, Adaptive block-structured mesh

FERREIRA, V. M. V., **Método híbrido LES / FDF Lagrangiana em malha adaptativa, bloco-estruturada.** 2015. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

Esta dissertação é parte de um amplo projeto de pesquisa, que visa ao desenvolvimento de uma plataforma computacional de dinâmica dos fluidos (CFD) capaz de simular a física de escoamentos que envolvem mistura de várias espécies químicas, com reação e combustão, utilizando um método hibrido Simulação de Grandes Escalas (LES) / Função Densidade Filtrada (FDF) Lagrangiana em malha adaptativa, bloco-estruturada. Uma vez que escoamentos com mistura proporcionam fenômenos que podem ser correlacionados com a combustão em escoamentos turbulentos, uma visão global da fenomenologia de mistura foi apresentada e escoamentos fechados, laminar e turbulento, que envolvem mistura de duas espécies químicas inicialmente segregadas foram simulados utilizando o código de desenvolvimento interno AMR3D e o código recentemente desenvolvido FDF Lagrangiana de composição. A primeira etapa deste trabalho consistiu na criação de um modelo computacional de partículas estocásticas em ambiente de processamento distribuído. Isto foi alcançado com a construção de um mapa Lagrangiano paralelo, que pode gerenciar diferentes tipos de elementos lagrangianos, incluindo partículas estocásticas, particulados, sensores e nós computacionais intrínsecos dos métodos Fronteira Imersa e Acompanhamento de Interface. O mapa conecta informações Lagrangianas com a plataforma Euleriana do código AMR3D, no qual equações de transporte são resolvidas. O método FDF Lagrangiana de composição realiza cálculos algébricos sobre partículas estocásticas e provê campos de composição estatisticamente equivalentes aos obtidos quando se utiliza o método de Diferenças Finitas para solução de equações diferenciais parciais; a técnica de Monte Carlo foi utilizada para resolver um sistema derivado de equações diferenciais estocásticas (SDE). Os resultados concordaram com os *benchmarks*, que são simulações baseadas em plataforma de Diferenças Finitas para solução de uma equação de transporte de composição filtrada.

Palavras-chave: Escoamento com mistura, Simulação de Grandes Escalas, Função Densidade Filtrada, Método de Monte Carlo, Malha adaptativa bloco-estruturada

List of Figures

List of Tables

Nomenclature

Abbreviations

AMR	-	Adaptive, block-structured mesh
CFD	-	Computational Fluid Dynamics
DNS	-	Direct Numerical Simulation
FDM	-	Finite Differences method
FDF	-	Filtered Density Function
IEM	-	Interaction by Exchange with the Mean micromixing model
LES	-	Large Eddy Simulation
MC	-	Monte Carlo method
MPI	-	Message Passing interface
NPC	-	Number of Particles per Cell
$PaSR$	-	Partially Stirred Reactor
PDE	-	Partial Differential Equations
PDF	-	Probability Density Function
RAS	-	Reynolds Averaged Simulation
SDE	-	Stochastic Differential Equations
SGS	-	Subgrid scale

Greek

α	-	chemical species
δ	-	Dirac delta
δ_{ij}	-	Kronecker delta
δ_m	-	initial mixture width
Δ	-	LES filter width
Δ^l	-	mesh length at AMR level l
Γ	-	molecular diffusivity
Γ_α	-	molecular diffusivity of the chemical species α
Γ^{SGS}	-	subgrid scale molecular diffusivity
μ	-	dynamic viscosity
μ^{SGS}	-	subgrid scale viscosity
ν	-	kinematic viscosity
$\mathrm{O}\left(n\right)$	-	big-Oh notation - upper bound of order n
ρ	-	density
$\left\langle\rho\right\rangle_\Delta$	-	filtered density
τ_{ij}	-	viscous stress tensor
τ_{ij}^{SGS}	-	subgrid scale viscous stress tensor
$\left\langle\tau_{ij}\right\rangle_\Delta$	-	filtered viscous stress tensor
ϕ	-	random variable - composition
ϕ^*	-	particle composition
$\overline{\phi^*}$	-	mean particle composition
ϕ_α	-	composition of chemical species α
$\left\langle\phi_\alpha\right\rangle_\Delta$	-	filtered composition of chemical species α
Φ	-	composition array
ψ_α	-	sample space of composition of chemical species α
Ψ	-	sample space of composition array
Ω_m	-	LES turbulent frequency
$\Omega\left(n\right)$	-	big-Omega notation - lower bound of order n
ξ_i	-	random variable with Gaussian distribution
ζ	-	composition field
$\overline{\zeta}$	-	overall composition mean field
$\left\langle\zeta\right\rangle$	-	temporal composition average

Latin

C_s	-	Smagorinsky constant
C_Ω	-	mechanical-to-scalar time-scale ratio
dW	-	increment of the Wiener process
f_ϕ	-	filtered density function of ϕ
F_ϕ	-	distribution function of ϕ
$f_{\Delta,\Phi}$	-	filtered density function of the composition array Φ
$f_{\Delta,u_i\Phi}$	-	joint velocity-composition filtered density function
G	-	spatial filter function
h_m	-	initial mixture height
$J_{\alpha j}$	-	molecular diffusion flux of the α chemical species in the j direction
$\langle J_{\alpha j}\rangle_\Delta$	-	filtered molecular diffusion flux
$J_{\alpha j}^{SGS}$	-	subgrid scale molecular diffusion flux
l_I	-	integral length scale
l_k	-	Kolmogorov length scale
Le	-	Lewis number
N_S	-	number of chemical species
p	-	pressure
$\langle p\rangle_\Delta$	-	filtered pressure
Q	-	generic function of a physical quantity
Q'	-	generic function fluctuation
$\langle Q\rangle_\Delta$	-	filtered generic function
Re	-	Reynolds number
S^2	-	composition variance based on temporal composition average
S_i^u	-	body forces in momentum transport equation
$\hat{S}_i^u$	-	filtered body forces
S_α	-	chemical reaction rate of the α chemical species
$\hat{S}_\alpha$	-	filtered chemical reaction
Sc	-	Schmidt number
Sc^{SGS}	-	subgrid Schmidt number
t	-	time
$u_i(x_i,t)$	-	velocity components
$\hat{u}_i$	-	filtered velocity - momentum transport notation
$\langle u_i\rangle_\Delta$	-	filtered velocity - FDF transport notation
V	-	volume of a grid cell
w^*	-	particle weight
X_i	-	spatial coordinates
Y_α	-	mass fractions of the chemical species α
$\hat{Y}_\alpha$	-	filtered mass fraction

Contents

CHAPTER I

INTRODUCTION

Multi-species mixing flows are everywhere: in a cup of tea, individual species mix with water until becoming a homogeneous fluid; in a car exhaust system and in an industry chimney, the hot combustion products outflow to atmosphere and mix with pure air; and in many industrial processes, fluids with different essences mix each other to create a specific product.

In order to simulate this kind of flows, we present a hybrid Large Eddy Simulation (LES) / Lagrangian composition Filtered Density Function (FDF) method on adaptive, block-structured mesh (AMR). The LES approach solves the large scales of the flow, while it models the small ones; on the other hand, the FDF method provides the chemical species composition fields using the Monte Carlo method, which solves a system of stochastic equations. Moreover, the AMR is both used to numerically solve the filtered transport equations and to represent the composition FDF using notional particles; this feature matches both LES and Monte Carlo method good practices i.e., to increase accuracy in regions of flow with more turbulent intensity and with sharp variations in composition field, respectively.

We focused on laminar and turbulent, non-reactive, two-species mixing flows with unit density and constant fluid properties. However, the described mathematical and numerical models are not restricted to those simplifications and may be used to solve a wide range of flows, such as reactive ones. In fact, that hybrid method has great advantage when applied to reactive and combustion flows, since it models in a simple way the chemical reaction term of species transport equations.

To that extent, this master thesis aims at exposing the development of the parallel Lagrangian composition FDF code and its merging with the in-house built AMR3D code. The FDF code performs algebraic calculations over an ensemble of notional particles using the Monte Carlo method and produces composition fields statistically equivalent to those obtained by Finite Differences numerical solution of partially differential equations (PDE). The complete Computational Fluid Dynamics (CFD) code allowed us to implement the before mentioned hybrid method and to simulate mixing flows.

The first step towards this objective consisted of building a computational model of notional particles on distributed processing environment. We achieved it constructing a parallel Lagrangian map, which can hold different types of Lagrangian elements, including notional particles. The map connects Lagrangian information with the Eulerian framework of the AMR3D code, in which transport equations are solved. We used Message Passing Interface (MPI) to communicate both Eulerian and Lagrangian data among processes.

The developed Lagrangian framework uses the multi-level hash table based on *Uthash*. It is built according to the Eulerian mesh. Uthash is a C header file which provides tools to create and manipulate hash tables, and is useful to quickly manage data.

We expose an overview of mixing phenomenology on free-shear layer flow, since it is a common studied flow, though we simulated non-compressible, lid-driven cavity, initially segregated mixing flows to verify the statistical equivalence between Eulerian and Lagrangian approaches for solving a composition field.

Moreover, we adopted the Schmidt number for all simulations $Sc = 0.7$; the Reynolds numbers $Re = 50$, 500 and 5000; the fluid density $\rho = 1.0$; the Lewis number $Le = 1.0$; and the mechanical-to-scalar time-scale ratio $C_\Omega = 2.0$.

We present results running a Intel® Core™ i5-3450, 3.10 GHz CPU, 1,600.00 MHz speed computer.

1.1 PDF methodology

Lagrangian Probability Density Function (PDF) methods originated in the eighties from the junction of Eulerian PDF methods and stochastic Lagrangian models. They consist in a feasible way to calculate the statistics of inhomogeneous turbulent flows (POPE, 1994).

An ensemble of notional particles represents a real statistical sample, wherein each notional particle stands for a single realization of the flow in space and time. The Lagrangian element has its own position, velocity and composition, that advance in time according to a system of stochastic differential equations (POPE, 1994). In the composition PDF method, the velocity comes from the Eulerian mesh by interpolation.

Peyman Givi suggested the FDF method in 1989 and Stephen Pope formulated it one year later to numerically solve turbulent reacting flows under Large Eddy Simulation approach (HAWORTH, 2010). Essentially, the FDF method is analogue to the PDF one in Reynolds-Averaged Simulations. It uses the probability density function of the subgrid scale scalar quantity to compute the effects of unresolved scalar fluctuations (COLUCCI et al., 1998).

1.2 Monte Carlo technique

The Lagrangian FDF method uses the Monte Carlo (MC) technique to solve a derived system of stochastic differential equations (COLUCCI et al., 1998). It works with ensembles of notional particles, which are statistical sampling of the expected field.

Stanislaw Ulam and John von Neumann create the MC method in the forties. But it was Nicolas Metropolis who named it, referring a story involving a famous casino in Monte Carlo. Additionally, Enrico Fermi — a physicist who helped to create modern physics — developed simultaneous and independently this statistical sampling technique. Figure (1.1) shows the analogical device created by Fermi — named FERMIAC — to study neutron transport via random choices of parameters.

Contemporary to the first electronic computer (ENIAC) and the first atomic bomb detonation test at Alamogordo, the Monte Carlo technique is still a powerful method that overcomes the ever present machine power-processing limitation. The ENIAC was essential

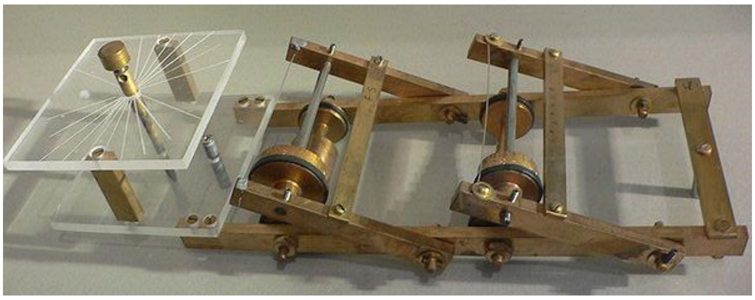


Figure 1.1: The FERMIAC device, http://en.wikipedia.org/wiki/FERMIAC, accessed on 18/04/2014.

to create MC method, since manual statistical calculations were no longer used because of tediousness.

Many problems are unfeasible when treated by Finite Differences methods — due to local-high gradient, for example. Instead, these are easily handled using MC. Indeed, this algorithm is recommended to evaluate complex and multiple-dimensional integrals. Its first scientific application was to solve the problem of neutron diffusion in fissionable materials, wherein von Neumann had increasingly become involved (METROPOLIS, 1987).

1.3 Mixing flows

We can classify a flow based on its physical aspects and patterns of turbulence transition. It may be a free shear flow, when it occurs without geometric restrictions e.g., a wake, a jet and a mixing layer; a boundary layer, when the flow interacts with a solid surface, forming an intense shear region; and an enclosed flow, with strong shear effects due to the presence of many walls (SILVEIRA-NETO, 2002).

Free-shear layer flows have a typical laminar-turbulence transition, starting with the Kelvin-Helmholtz instabilities. They are usually found in diffusive combustion phenomena due to the presence of two streams with different velocities, which in diffusive combustion also present two segregate fluids, e.g., jet fire in a pipeline leakage. One can find enclosed flows into chemical reactors and inside pipelines, for example. A mix of these two kinds of flows occurs inside a furnace; it is composed by a jet fire that outflows to a chamber, where hot gases circulate.

Mixing flows occur frequently in nature and in a variety of engineering processes, specially in turbulent regime. Among their effects, it may be cited the change in compound concentrations in rivers (BOUCHEZ et al., 2010), the combustion efficiency in engines (SOUZA; OLIVEIRA, 2010) and the losses of millions per year associated to insufficient mixing in chemical reactors (PAUL; ATIEMO-OBERG; KRESTA, 2004). Figure (1.2) shows a numerical simulation of burning gases that outflow a swirled combustor.

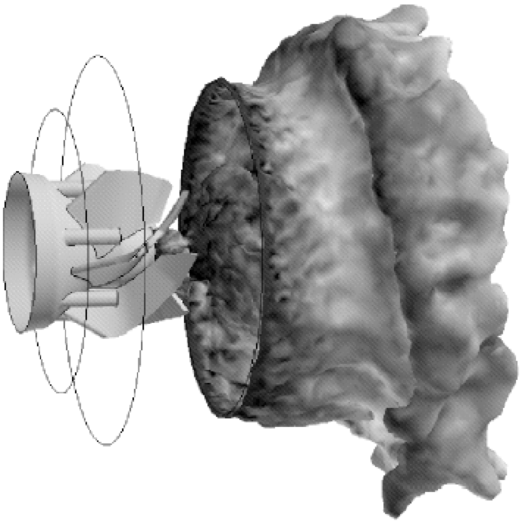


Figure 1.2: Instantaneous temperature field — isosurface of 1100 K. Large Eddy Simulation of a swirled combustor during turbulent combustion (POINSOT; VEYNANTE, 2005).

Despite the origin of turbulence study is contemporary to Osborn Reynolds, with his experiment of 1883, scientific knowledge on mixing was established in the second half of the 20$^{\text{th}}$ century — highlighting the publication by Nagata in 1975. Henceforth, the understanding of mixing phenomena has allowed many engineering projects (PAUL; ATIEMO-OBERG; KRESTA, 2004).

As example of a turbulent mixing flow in nature, the work of Bouchez et al. (2010) covers the sodium distribution in Solimões river — downstream its junction with Purús river. Its objective was to verify the water perfect mixture hypothesis Fig. (1.3).

Mixing is present in many industrial applications. It can be defined as the reduction of composition, phases and/or temperature heterogeneity, aiming to achieve a desired product. Among the industries which large losses because of mixing inefficiency in the eighties and nineties are the chemical and pharmaceutical industries. They lost up to 10 billions per year because of poor mixing (PAUL; ATIEMO-OBERG; KRESTA, 2004).

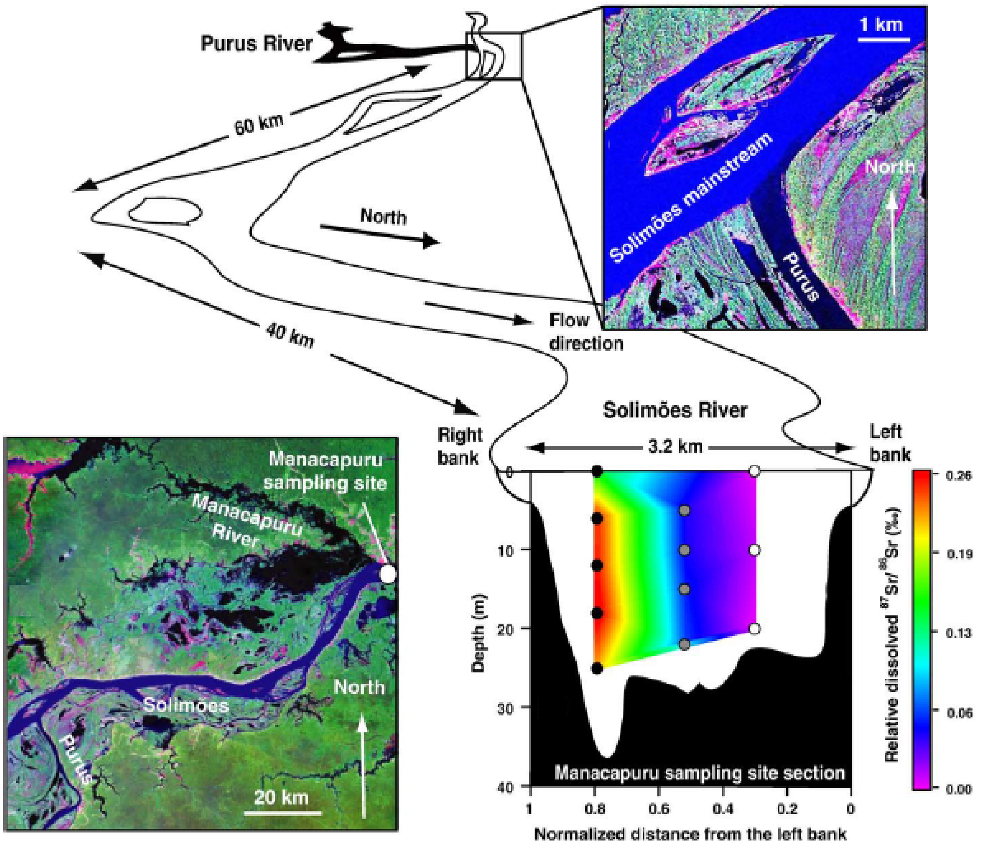


Figure 1.3: Aerial view of the junction of Purús and Solimões rivers; aerial view of the course of Solimões river from its junction with Purús river to Manacapuru metering site; and transversal section of Solimões river showing depth profile and compounds concentration field (BOUCHEZ et al., 2010).

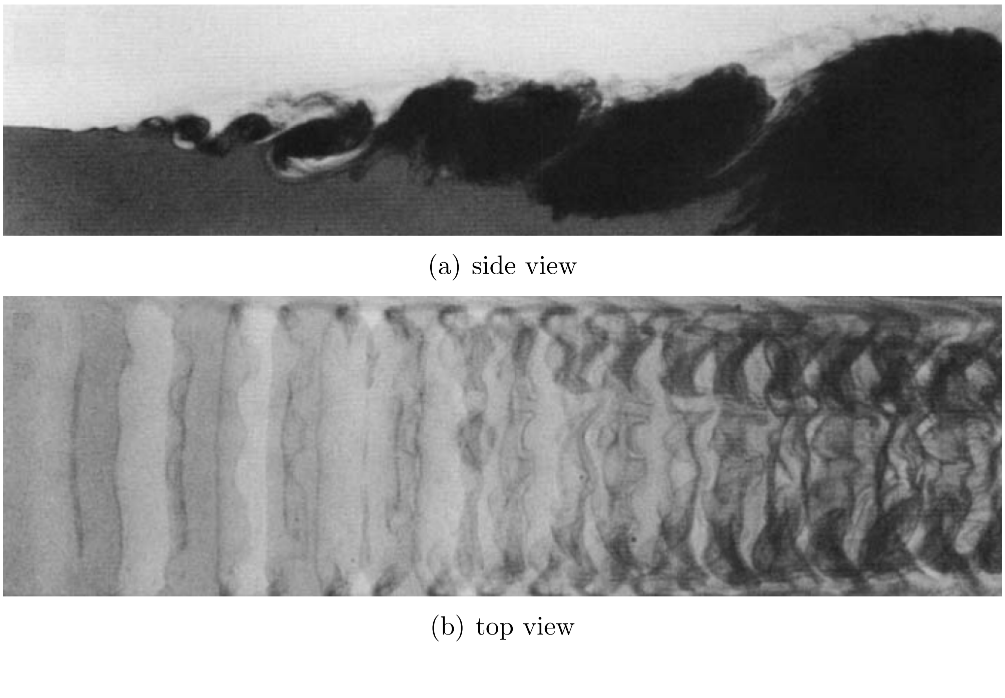
1.3.1 *Phenomenology*

Turbulent mixing on shear flows has been subject of scientific research since the 19th century. The work of Konrad (1977) in the seventies is one among many that have been developed in the 20th century. During the eighties, theories experienced a breakthrough (BREIDENTHAL, 1981; BROADWELL; BREIDENTHAL, 1982; KOOCHESFAHANI; DIMOTAKIS, 1986). In the nineties, an increase of the capacity and accuracy of experimental techniques enabled a deep view into the fine-scale structures of the scalar field (BUCH; DAHM, 1996, 1998).

The early questions that arose in the research of multiple-species, turbulent shear layer flows were regarding the molecular mixing effect and the influence of the Reynolds number on mixing (KONRAD, 1977). Konrad (1977) used a concentration probe with frequency response of $100\,kHz$ and spatial resolution of $0.1\,mm$ to study the mixture extent in these flows. The experiment had two parallel streams of high speed helium and low speed nitrogen. The author also analyzed streams with equal density fluids. The results revealed strong Reynolds number influence on mixing, which increased up to $25\,\%$ with the increase in Reynolds number.

Because species diffusion is the last step during mixing, several authors became interested in discovering the influence of the Schmidt number (Sc) on multiple-species, turbulent shear layer flows, such as Breidenthal (1981).

The Schmidt number $Sc = \nu/\Gamma$ is a dimensionless parameter that expresses the relative magnitude between the momentum $\nu = \mu/\rho$ and species Γ diffusivities in a fluid. Liquids and gases usually have $Sc \gg 1$ and $Sc \approx 1$, respectively. Since diffusion works towards homogenizing the momentum and the composition fields, the smallest composition structure — named Batchelor scale — may be even smaller than the momentum structure — named Kolmogorov scale — in a liquid turbulent flow (FOX, 2003).

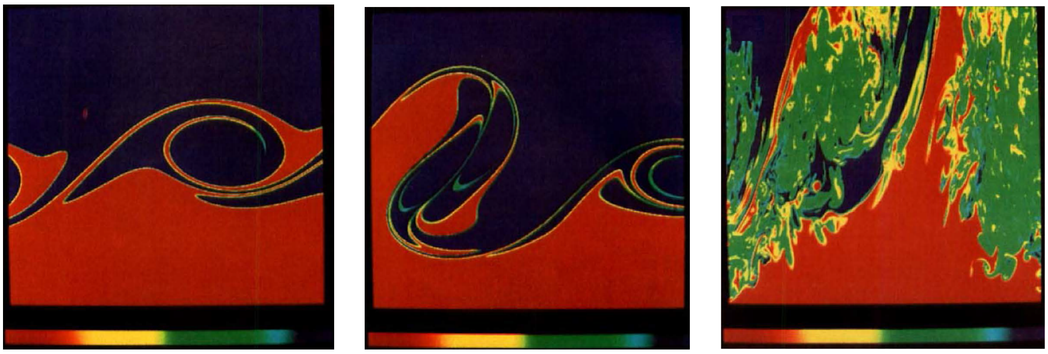


(a) side view

(b) top view

Figure 1.4: Reacting, turbulent shear layer flow. The product of reaction is dark-grey colored (here in gray color scales); it locates around the large structures of the flow (BREIDENTHAL, 1981).

Breidenthal (1981) built a turbulent flow experiment which used water-diluted chemical reactants ($Sc = 600$) to measure the extent of molecular mixing. Figure (1.4) shows the visible product of reaction.

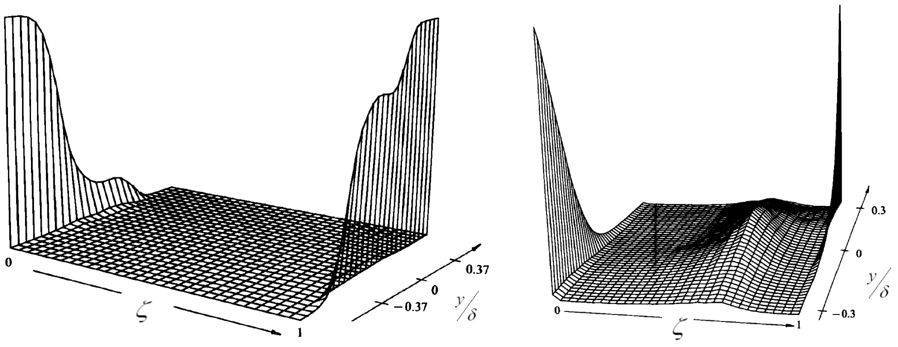


(a) (b) (c)

Figure 1.5: Concentration field — red: low-speed, black: high-speed, and the remaining colors: mixed fluid (KOOCHESFAHANI; DIMOTAKIS, 1986).

Broadwell and Breidenthal (1982) observed two remaining unmixed streams in coherent flow structures and proposed a model for mixing on multiple-species, turbulent shear layer flows.

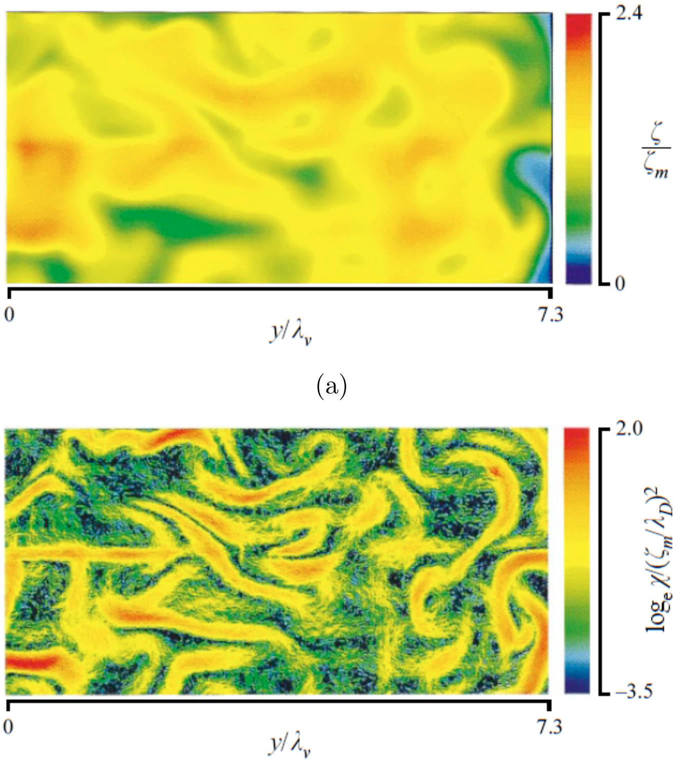
Koochesfahani and Dimotakis (1986) presented an experiment of mixing in turbulent shear layer flows with large Schmidt number. The study covered two cases: reacting flows, to directly measure mixing extent, and non-reacting flows, to calculate the PDF of a passive scalar.



(a) Below mixture transition (b) Above mixture transition

Figure 1.6: PDF of composition field. The local thickness δ in the y/δ axis corresponds to 1% of the total mixed-fluid probability (KOOCHESFAHANI; DIMOTAKIS, 1986).

Figures (1.5) and (1.6) show that the cores of vortex structures are mainly composed by fluid from high-speed stream. In the range of mixture transition, which starts approximately at $Re = 5600$, mixture occurs spatially and also in terms of composition change. The maximum and minimum fluid composition varies slightly during the transversal extent of the turbulent shear layer for $Re \geq 5600$. For flows above mixture transition, the probability of finding an unmixed fluid in the center of the layer is 0.45. Furthermore, the mean concentration is approximately constant throughout the layer (KOOCHESFAHANI; DIMOTAKIS, 1986).



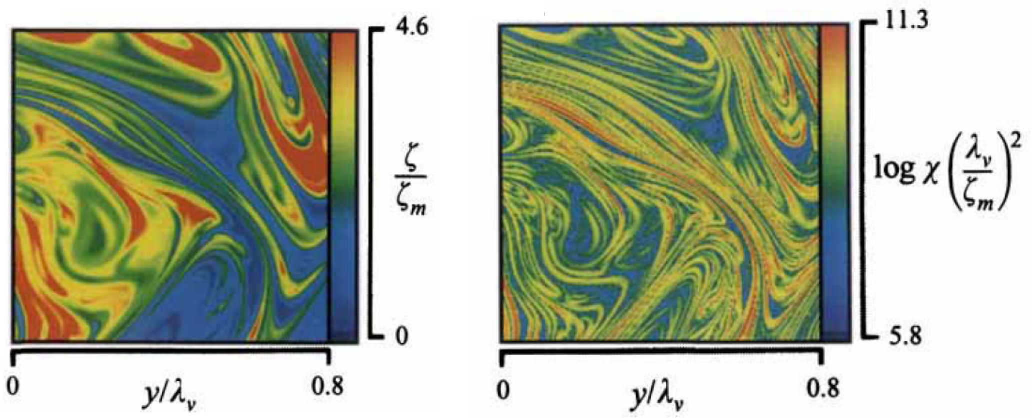
(a)

(b)

Figure 1.7: (a) Scalar and (b) scalar dissipation rate fields for $Sc \approx 1$ (BUCH; DAHM, 1998).

Koochesfahani and Dimotakis (1986) proved the Schmidt number has great influence over mixing. At high local Reynolds numbers, the normalized amount of chemical product formed in liquid is 50 % less than in gas flows. Since molecular diffusion occurs at Batchelor scale and liquid flows spend more time to achieve these scales than gas flows, reaction occurs at lower rate in liquid than in gas flows.

Koochesfahani and Dimotakis (1986) produced the spatial-time picture Fig. (1.5) by successive scans that mapped the transverse coordinate of the shear layer with spatial and temporal resolution of $10\, points/mm$ and $1.25\, scans/ms$. Figure (1.5) shows poor mixing is the result of few interface regions between two streams. A green color represents the predominant concentration of mixed fluid.



(a)

(b)

Figure 1.8: (a) Scalar and (b) scalar dissipation rate fields for $Sc \approx 2000$ (BUCH; DAHM, 1996).

Figure (1.6) shows two delta functions at $\zeta = 0$ and $\zeta = 1$, revealing the predominance of unmixed fluid in shear layer. The rich high speed fluid ($\zeta = 1$) in the cores of vortex structures is highlighted in y/δ axis. Figure (1.6) also shows the fluid composition changes towards the mean concentration ($\zeta = 0.6$), shaping a new delta function.

The multiple-species, turbulent shear flows present a similar fine-scale structure of scalar dissipation rate field at high and nearly unit Schmidt number. This structure is formed by strained, laminar and sheet-like diffusion layers (BUCH; DAHM, 1998). Buch and Dahm (1996, 1998) investigated the composition field in turbulent shear flows at $Sc \approx 2000$ and $Sc \approx 1$. The resolution and signal quality of measurements led to obtain the scalar and the instantaneous dissipation rate of scalar energy fields.

Figures (1.7) and (1.8) show the scalar and the scalar dissipation rate fields for gas and liquid flows, respectively. They highlight the fine-scale similarity between these two flows, and reveal liquid flows have scalar wave-lengths smaller than gas flows.

We have exposed an overview of mixing phenomenology on free-shear layer mixing flows from large to small turbulence scales. It grants us understanding on the physics of the flow and support us to apply mathematical models that agree with these physics e.g., LES approach, subgrid modeling and closures on composition FDF transport equation.

In this way, the experiments presented in this section were neither simulated nor had their results compared against numerical data; instead, our simulations covered just enclosed, mixing flows. Even though, the general relation between Reynolds number and mixing extent described in this section could be observed on our results.

The next chapter shows the mathematical modeling for the multiple-species mixing flows we have described. Section 2.1 presents the transport equations of mass, momentum and chemical species' mass fraction with the closures for the viscous stress tensor and the molecular diffusion flux. Section 2.2 covers the Large Eddy Simulation approach which separates turbulent flow scales in order to numerically solve the large scales and model the small ones. We show the filtered density function concept and the composition FDF transport equation in section 2.3. The derived system of stochastic differential equations for composition transport appears in section 2.4.

Chapter 3 shows the Lagrangian map and the overall computational framework developed to support the Lagrangian composition FDF code. It also present results concerning particles transport in parallel environment. Section 3.1 explains what is an adaptive, block-structured mesh to which the Lagrangian map must conform. Section 3.2 shows how the link between the Eulerian and Lagrangian frameworks was achieved. We cover the Lagrangian distributed processing in section 3.3, showing some of the Message Passing Interface terms and functions, as well as the particle identifiers controller.

Chapter 4 exposes the developed Lagrangian composition FDF code, which communicates with the Finite Differences framework of the AMR3D code. Section 4.2 describes the Lagrangian Monte Carlo algorithm that is applied upon the system of stochastic differential equations to solve the composition field of the flow. Section 4.3 covers the central point of MC solver: the population of notional particles constraint. We present the Partially Stirred Reactor concept, some mathematical models and a single box PaSR simulation in section 4.4. The statistical equivalence between Eulerian and Lagrangian approaches appears in section 4.5; this section also presents simulations of enclosed, initially segregated multiple-species mixture flows for laminar and turbulent states.

CHAPTER II

MATHEMATICAL MODELING

This chapter describes the hybrid Large Eddy Simulation (LES) / Lagrangian Filtered Density Function (FDF) mathematical model implemented in the AMR3D code (VILLAR, 2007; LIMA, 2012). Section 2.1 details the transport equations of mass (continuity), momentum (Navier-Stokes) and composition; section 2.2 describes the LES approach used to filter these equations, as well as the closure models required for numerical solutions; sections 2.3 and 2.4 refer to the FDF transport equations and to the system of derived stochastic differential equations (SDE).

2.1 Transport equations

The transport equations for reactive flows, with velocity vector $u_i\left(x_i,t\right)$ $(i=1,2,3)$, density $\rho\left(x_i,t\right)$ and chemical species mass fraction array $Y_\alpha\left(x_i,t\right)$ $(\alpha=1,2,...,Ns)$ is spatially, x_i, and temporally, t, dependent

$$\frac{\partial\rho}{\partial t}+\frac{\partial\rho u_i}{\partial x_i}=0, \tag{2.1}$$

$$\frac{\partial\rho u_i}{\partial t}+\frac{\partial\rho u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{\partial\tau_{ij}}{\partial x_j}+S_i^u, \tag{2.2}$$

$$\frac{\partial\rho Y_\alpha}{\partial t}+\frac{\partial\rho u_j Y_\alpha}{\partial x_j}=-\frac{\partial J_{\alpha j}}{\partial x_j}+S_\alpha, \tag{2.3}$$

where: τ_{ij}, S^u_i, S_α and $J_{\alpha j}$ stand for viscous stress tensor, body forces, chemical reaction rate and molecular diffusion flux for species α, respectively; i and j are directions in which summation apply; p is the pressure.

For a Newtonian fluid, the viscous stress tensor is modeled as

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, \tag{2.4}$$

where, μ is the fluid molecular viscosity and δ_{ij} is the delta Kronecker tensor.

The molecular diffusion flux $J_{\alpha j}$ is modeled using Fick's law of diffusion

$$J_{\alpha j} = -\frac{\mu}{Sc} \frac{\partial Y_\alpha}{\partial x_j}, \tag{2.5}$$

where: $Sc = \nu / \Gamma_\alpha$ is the Schmidt number; ν and Γ_α are the kinematic viscosity and molecular diffusivity for species α, respectively. The following equations assume an equal molecular diffusivity to all species, $\Gamma_\alpha = \Gamma$, and a unit Lewis number, $Le = 1$.

When a flow comprises thermal energy exchange, it is useful to group Y_α from Eq. (2.3) and the enthalpy into a single generic scalar variable, ϕ_α. The transport equation of that generic scalar becomes

$$\frac{\partial \rho \phi_\alpha}{\partial t} + \frac{\partial \rho u_j \phi_\alpha}{\partial x_j} = -\frac{\partial J_{\alpha j}}{\partial x_j} + S_\alpha, \tag{2.6}$$

with

$$J_{\alpha j} = -\rho \Gamma \frac{\partial \phi_\alpha}{\partial x_j}. \tag{2.7}$$

Vedovoto (2011) presented more details about the enthalpy transport and the generic scalar modeling.

At this point one could numerically solve the presented $Ns + 3$ transport equations Eqs. (2.1) to (2.3) with their closures Eqs. (2.4) and (2.5) — just $Ns - 1$ equations must be solved in order to get all Ns species, because $\sum\limits_{\alpha=1}^{Ns} Y_\alpha = 1$. However, as the number of transported chemical species increases the computational resources required to solve this set of equations also increases. Additionally, all flow's degrees of freedom must be captured by the computational mesh on Finite Differences or Finite Volume approaches. Both scenarios may be impracticable for high Reynolds number flows and nowadays available computational resources.

We introduce a hybrid LES / Lagrangian FDF method which is an alternative approach to numerically solve mixing flows, specially for reactive and combustion ones. The FDF methodology greatest advantage is to present the chemical reaction rate term in a closed form, free from modeling efforts needed in Finite Differences method (VEDOVOTO, 2011).

2.2 Large Eddy Simulation

We can divide the wavelengths of the flow into large (resolved) and small (modeled) scales — captured and not captured by the computational mesh, respectively. This strategy is called Large Eddy Simulation in computational study of turbulent flows (SILVEIRA-NETO, 2002). Direct Numerical Simulation (DNS) and Reynolds Averaged Simulation (RAS) are alternative approaches. The former solves every scale of the flow, while the latter solves the mean field.

In order to choose the best approach to numerically solve the mathematical model of a given flow, a question must be answered: how much the large and the small scales affect the flow? In a turbulent free shear flow, the large scales control the flow dynamics. Molecular mixing occurs at the smallest scales — regarding space and time — and determines how momentum and species concentration fields evolve (HAWORTH, 2010).

We present the spatial filtering operation over the mathematical model developed in section 2.1. Given Q any function of a physical quantity (e.g., u_i or Y_α), its local spatially filtered value $\langle Q\rangle_\Delta$ is defined as the integral over the flow domain

$$\langle Q\rangle_\Delta = \langle Q\left(x_i, t\right)\rangle_\Delta \equiv \int Q\left(y_i, t\right) G\left(\left|y_i - x_i\right|\right) dy, \tag{2.8}$$

where, Δ is the characteristic filter width and $G\left(z_i\right)$ is the spatial filter function that satisfies

$\int\limits_{-\infty}^{+\infty} G\left(z_i\right) dz = 1$ for $z_i = |y_i - x_i|$,

$$G\left(z_i\right) = \begin{cases} 1/\Delta^3 & if\, z_i \leq \Delta, \qquad i = 1,2,3 \\ \\ 0 & otherwise \end{cases}, \tag{2.9}$$

where, $\Delta = V^{1/3}$ and V is the cell volume.

Therefore, the instantaneous value of $Q\left(x_i, t\right)$ can be decomposed into its filtered $\langle Q\left(x_i, t\right)\rangle_\Delta$ and fluctuation $Q'\left(x_i, t\right)$ parts. Two properties of filters are (SILVEIRA-NETO, 2002):

- To filter a filtered quantity affects its value

$$\langle\langle Q\rangle_\Delta\rangle_\Delta \neq \langle Q\rangle_\Delta; \tag{2.10}$$

- The filter of a fluctuation is not equal zero

$$\left\langle Q'\right\rangle_\Delta \neq 0. \tag{2.11}$$

Applying these properties on the set of equations (2.1 to 2.3) we have,

$$\frac{\partial\langle\rho\rangle_\Delta}{\partial t} + \frac{\partial\langle\rho\rangle_\Delta \widehat{u}_i}{\partial x_i} = 0, \tag{2.12}$$

$$\frac{\partial\langle\rho\rangle_\Delta \widehat{u}_i}{\partial t} + \frac{\partial\langle\rho\rangle_\Delta \widehat{u}_i \widehat{u}_j}{\partial x_j} = -\frac{\partial\langle p\rangle_\Delta}{\partial x_i} + \frac{\partial\langle\tau_{ij}\rangle_\Delta}{\partial x_j} + \frac{\partial\tau_{ij}^{SGS}}{\partial x_j} + \widehat{S}_i^u, \tag{2.13}$$

$$\frac{\partial\langle\rho\rangle_\Delta \widehat{Y}_\alpha}{\partial t} + \frac{\partial\langle\rho\rangle_\Delta \widehat{u}_j \widehat{Y}_\alpha}{\partial x_j} = -\frac{\partial\langle J_{\alpha j}\rangle_\Delta}{\partial x_j} + \frac{\partial J_{\alpha j}^{SGS}}{\partial x_j} + \widehat{S}_\alpha, \tag{2.14}$$

where: $\langle\rangle_\Delta$ and $\widehat{}$ are both filter notations equivalently used throughout this chapter; $\tau_{ij}^{SGS} = \langle\rho\rangle_\Delta \widehat{u}_i \widehat{u}_j - \langle\rho\rangle_\Delta \widehat{u_i u_j}$ and $J_{\alpha j}^{SGS} = \langle\rho\rangle_\Delta \widehat{u}_j \widehat{Y}_\alpha - \langle\rho\rangle_\Delta \widehat{u_j Y_\alpha}$ are the subgrid scale (SGS) stress and the SGS molecular diffusion flux, respectively.

Turbulent mixing analysis requires to establish models for τ_{ij}^{SGS} and $J_{\alpha j}^{SGS}$ in order to close the system of filtered equations. It occurs because the filtered product of two instantaneous variables are unknown (e.g., $\widehat{u_i u_j}$ and $\widehat{u_j Y_\alpha}$).

In reacting flows, Eq. (2.14) demands a model for the filtered chemical reaction term. The FDF formulation attends this requeriment presenting this term in a closed form. It is the main benefit of the FDF schemes (COLUCCI et al., 1998), since the filtered chemical reaction term modeling is difficult due to the non-linear relationship between chemical reaction and turbulence (FOX, 2003). Fox (2003) presents alternative approaches for modeling the chemical reaction term for LES and RAS perspectives.

The key point for modeling whatever SGS quantity is its relationship with a known large scale. In the case of the unknown SGS stress, the Smagorinsky eddy viscosity closure may relate this term with the local large scale rate of flow strain (COLUCCI et al., 1998).

By applying the Smagorinsky model, Eq. (2.13) incorporates a SGS viscosity μ^{SGS} in the filtered stress term. The modeled filtered stress is represented by

$$\langle \tau_{ij} \rangle_{\Delta}^{M} = \left(\mu + \mu^{SGS}\right) \left[\left(\frac{\partial \widehat{u}_i}{\partial x_j} + \frac{\partial \widehat{u}_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial \widehat{u}_k}{\partial x_k} \delta_{ij} \right], \tag{2.15}$$

where the SGS viscosity — also called apparent or turbulent viscosity — is proportional to the filtered strain rate $\widehat{S}_{ij}$,

$$\mu^{SGS} = (Cs\Delta)^2 \langle \rho \rangle_{\Delta} \left| \widehat{S} \right|, \tag{2.16}$$

$$\left| \widehat{S} \right| = \left(2 \widehat{S}_{ij} \widehat{S}_{ij} \right)^{1/2}, \tag{2.17}$$

$$\widehat{S}_{ij} = \frac{1}{2} \left(\frac{\partial \widehat{u}_i}{\partial x_j} + \frac{\partial \widehat{u}_j}{\partial x_i} \right), \tag{2.18}$$

and C_s is the Smagorinsky constant, which value 0.2 is used for isotropic turbulence (FERZIGER; PERIC, 1996).

The gradient diffusion concept relates the unknown SGS molecular diffusion flux with the known filtered species' mass fraction (FOX, 2003) — a similar approach used by applying Fick's law in Eq. (2.5) —

$$J_{\alpha j}^{SGS} = -\langle \rho \rangle_{\Delta} \Gamma^{SGS} \frac{\partial \widehat{Y}_\alpha}{\partial x_j}, \tag{2.19}$$

where,

$$\Gamma^{SGS} = \frac{2\langle\rho\rangle_{\Delta}(Cs\Delta)^2}{Sc^{SGS}}\left|\widehat{S}\right| = \frac{\mu^{SGS}}{Sc^{SGS}}, \tag{2.20}$$

Γ^{SGS} is the SGS molecular diffusivity and Sc^{SGS} is the SGS Schmidt number, which controls the magnitude of turbulent molecular diffusion (VEDOVOTO, 2011).

The filtered transport equations Eqs. (2.12) to (2.14), the closure models Eqs. (2.15) to (2.20) and a chemical kinetics mechanism are sufficient to solve reactive and low Mach number flows.

Although the mathematical model presented in this chapter is in accordance with the flows of interest (i.e., variable density, multi-species reactive and low Mach number flows), the developments and the simulations presented in this master thesis are focused on two-species, non-reactive and incompressible (unit density fluid) flows.

2.3 Filtered Density Function

The Filtered Density Function is the Probability Density Function (PDF) of SGS variables (COLUCCI et al., 1998). Whilst the PDF length scales are restricted to the turbulence integral scale l_T, the FDF length scales fulfill a range down to the characteristic filter width (HAWORTH, 2010).

The FDF of a random variable ϕ (e.g., composition) is defined as the derivative of its distribution function with respect to its sample space ψ (i.e., its domain)

$$f_{\phi}\left(\psi\right) \equiv \frac{dF_{\phi}\left(\psi\right)}{d\psi}, \tag{2.21}$$

and the distribution function is defined as the probability of that variable be lesser than φ,

$$F_{\phi}\left(\psi\right) \equiv P\left(\phi < \varphi\right). \tag{2.22}$$

Therefore, the probability of Φ belongs to an infinitesimal interval in the sample space

Ψ defines the FDF at point x_i and time t (FOX, 2003),

$$f_{\Delta,\Phi}\left(\Psi;x_i,t\right) \equiv Prob\left\{\left(\Psi < \Phi\left(x_i,t\right) < \Psi + d\Psi\right)\right\}, \tag{2.23}$$

where, Φ is the random variable array and Ψ is its sample space.

Considering Φ the array of chemical species mass fractions Y_α, the composition FDF can be expressed as the integral of the Dirac delta function δ and the spatial filter G from Eq. (2.9),

$$f_{\Delta,\Phi}\left(\Psi;x_i,t\right) \equiv \int \delta\left(\Psi - \Phi\left(y_i,t\right)\right) G\left(y_i - x_i\right) dy, \tag{2.24}$$

where the integral covers the flow domain, and

$$\delta\left[\Psi - \Phi\left(y_i,t\right)\right] \equiv \prod_{\alpha=1}^{Ns} \delta\left[\psi_\alpha - \phi_\alpha\left(y_i,t\right)\right]. \tag{2.25}$$

Since $\delta\left[\Psi - \Phi\left(y_i,t\right)\right]$ is the detailed, low-level density, the composition FDF can be understood as the G filtered density (COLUCCI et al., 1998).

We present two ways for deriving the composition FDF transport equation. The first starts from the transport equations (2.2) and (2.3) and further integrates the joint velocity-composition FDF on velocity space in order to generate the composition FDF (section 2.3.1); the second starts from the time derivative of Eq. (2.24) (section 2.3.2).

2.3.1 Joint velocity-composition FDF

The joint velocity-composition FDF transport equations are an alternative mathematical model for the physical system being treated in this master thesis. This model considers both the momentum and composition fields in a probabilistic manner.

The probability of the random velocity vector u_i and the random species mass fraction array Φ belong to a range in the sample spaces v_i and Ψ, respectively, and defines the joint velocity-composition FDF at point x_i and time t (FOX, 2003)

$$\begin{aligned} f_{\Delta,u_i\Phi}\left(v_i,\Psi;x_i,t\right) &\equiv Prob\left\{\left(v_i < u_i\left(x_i,t\right) < v_i + dv_i\right)\right\} \cap \\ & \quad Prob\left\{\left(\Psi < \Phi\left(x_i,t\right) < \Psi + d\Psi\right)\right\}. \end{aligned} \tag{2.26}$$

The joint FDF is valuable because we can use it to derive all one-point statistics of u_i and Φ, for example the filtered velocity $\widehat{u}_i$, the filtered composition $\widehat{\phi}_\alpha$, the Reynolds stresses $\widehat{u_i u_j}$, the composition fluxes $\widehat{u}_i\phi_\alpha^{'}$ and the filtered chemical reaction rate $\widehat{S}_\alpha$. However, transported FDF methods need closures for two-point dependent terms; therefore, pressure fluctuation gradient, viscous and species diffusion need additional modeling (FOX, 2003).

In order to derive the joint FDF transport equation from Eqs. (2.2) and (2.3), two independent expressions for the filtered total derivative of an arbitrary, one-point scalar function of velocity and composition $Q\left(u_i,\Phi\right)$ must be found and further equated (FOX, 2003).

The filtered value of $Q\left(u_i,\Phi\right)$ is related with the joint FDF by

$$\left\langle Q\left(u_i,\Phi\right)\right\rangle_\Delta = \int\int Q\left(v_i,\Psi\right) f_{\Delta,u_i\Phi}\left(v_i,\Psi;x_i,t\right) dv d\Psi, \tag{2.27}$$

where the integrals cover all v_i and Ψ domains.

Firstly, we rewrite the transport equations in a compact and simplified form

$$\frac{Du_i}{Dt} \equiv \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = A_i, \tag{2.28}$$

$$\frac{D\phi_\alpha}{Dt} \equiv \frac{\partial \phi_\alpha}{\partial t} + \frac{\partial u_j \phi_\alpha}{\partial x_j} = \Theta_\alpha, \tag{2.29}$$

where,

$$A_i \equiv -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + S_i^u, \tag{2.30}$$

$$\Theta_\alpha \equiv \frac{\partial J_{\alpha j}}{\partial x_j} + S_\alpha. \tag{2.31}$$

Based on the linear nature of the derivative operator, the filtered total derivative of

$Q\left(u_i, \Phi\right)$ can be expressed as

$$\left\langle \frac{DQ}{Dt} \right\rangle_{\Delta} = \frac{\partial \left\langle Q \right\rangle_{\Delta}}{\partial t} + \frac{\partial \left\langle u_j Q \right\rangle_{\Delta}}{\partial x_j}. \tag{2.32}$$

Combining Eq. (2.32) with Eq. (2.27), and since v_i and Ψ are integration variables, it becomes

$$\left\langle \frac{DQ}{Dt} \right\rangle_{\Delta} = \int \int Q\left(v_i, \Psi\right) \left\{ \frac{\partial f_{\Delta,u_i\Phi}}{\partial t} + v_j \frac{\partial f_{\Delta,u_i\Phi}}{\partial x_j} \right\} dvd\Psi. \tag{2.33}$$

The chain rule is the starting point to derive the second form of the filtered total derivative of $Q\left(u_i, \Phi\right)$,

$$\frac{DQ}{Dt} = \frac{\partial Q}{\partial u_j} \frac{Du_i}{Dt} + \frac{\partial Q}{\partial \phi_\alpha} \frac{D\phi_\alpha}{Dt}. \tag{2.34}$$

Furthermore, substituting Eqs. (2.28) and (2.29) into Eq. (2.34) and applying the filter, we have

$$\left\langle \frac{DQ}{Dt} \right\rangle_{\Delta} = \left\langle \frac{\partial Q}{\partial u_j} A_i \right\rangle_{\Delta} + \left\langle \frac{\partial Q}{\partial \phi_\alpha} \Theta_\alpha \right\rangle_{\Delta}, \tag{2.35}$$

in which the conditional filter — see Fox (2003) — is considered to give

$$\left\langle \frac{DQ}{Dt} \right\rangle_{\Delta} = -\int \int Q\left(v_i, \Psi\right) \left\{ \frac{\partial}{\partial v_j} \left[\left\langle A_i | v_i, \Psi \right\rangle f_{\Delta,u_i\Phi} \right] + \frac{\partial}{\partial \psi_\alpha} \left[\left\langle \Theta_\alpha | v_i, \Psi \right\rangle f_{\Delta,u_i\Phi} \right] \right\} dvd\Psi. \tag{2.36}$$

Finally, the final form of the joint FDF transport equation is achieved by subtracting Eq. (2.36) from Eq. (2.33)

$$\frac{\partial f_{\Delta,u_i\Phi}}{\partial t} + \frac{\partial u_j f_{\Delta,u_i\Phi}}{\partial x_j} = -\frac{\partial}{\partial v_j} \left[\left\langle A_i | v_i, \Psi \right\rangle_{\Delta} f_{\Delta,u_i\phi} \right] - \frac{\partial}{\partial \psi_\alpha} \left[\left\langle \Theta_\alpha | v_i, \Psi \right\rangle_{\Delta} f_{\Delta,u_i\Phi} \right]. \tag{2.37}$$

Since the joint velocity-composition FDF transport equation has random variables describing three velocity components u_i and the array of chemical species Φ, when we integrate Eq. (2.37) over the velocity sample space v_i, it results the composition FDF transport equation Eq. (2.45).

Because this integration eliminates the velocity field, it is necessary to provide the momentum information and to treat the turbulence through a different model. Therefore, the composition and velocity fluctuations link no longer exists, and a FDF composition flux model is consequently required. For both joint FDF and composition FDF models, a micromixing model is necessary to calculate the molecular diffusion response on both FDF shape and rate of composition variance decay (FOX, 2003).

As mentioned before, we present in section 2.3.2 a second form for deriving the composition FDF transport equation. It starts from the composition FDF definition as the integral of the Dirac delta function δ and the spatial filter G from Eq. (2.9).

2.3.2 Composition FDF

The time derivative of Eq. (2.24) is

$$\begin{aligned}\frac{\partial f_{\Delta,\Phi}\left(\Psi;x_i,t\right)}{\partial t} &= -\int \frac{\partial \phi_\alpha\left(y_i,t\right)}{\partial t}\frac{\partial \delta\left[\Psi-\Phi\left(y_i,t\right)\right]}{\partial \psi_\alpha} \times G\left(y_i-x_i\right)dy \\ &= -\frac{\partial}{\partial \psi_\alpha}\int \frac{\partial \phi_\alpha\left(y_i,t\right)}{\partial t} \times \delta\left[\Psi-\Phi\left(y_i,t\right)\right]G\left(y_i-x_i\right)dy. \end{aligned} \tag{2.38}$$

Considering the conditional filter of $Q\left(x_i,t\right)$, which is any function of Y_α,

$$\left\langle Q\left(x_i,t\right) \mid \Psi\right\rangle_\Delta \equiv \frac{\int Q\left(y_i,t\right)\delta\left[\Psi-\Phi\left(x_i,t\right)\right]G\left(y_i-x_i\right)dy}{f_{\Delta,\Phi}\left(\Psi;x_i,t\right)}. \tag{2.39}$$

When $Q\left(x_i,t\right)$ is the partial, time derivative of $\phi_\alpha\left(x_i,t\right)$, Eq. (2.38) becomes

$$\frac{\partial f_{\Delta,\Phi}\left(\Psi;x_i,t\right)}{\partial t} = -\frac{\partial}{\partial \psi_\alpha}\left[\left\langle \frac{\partial \phi_\alpha}{\partial t}\,\middle|\,\Psi\right\rangle_\Delta f_{\Delta,\Phi}\left(\Psi;x_i,t\right)\right]. \tag{2.40}$$

For a unit density flow, Eq. (2.3) can be simplified. Substituting the reduced form of Eq. (2.3) in Eq. (2.40), we have

$$\begin{aligned}\frac{\partial f_{\Delta,\Phi}\left(\Psi;x_i,t\right)}{\partial t} &= \frac{\partial}{\partial \psi_\alpha}\left[\left\langle \frac{\partial u_j\phi_\alpha}{\partial x_j}\,\middle|\,\Psi\right\rangle_\Delta f_{\Delta,\Phi}\left(\Psi;x_i,t\right)\right] \\ &\quad +\frac{\partial}{\partial \psi_\alpha}\left[\left\langle \frac{\partial J_{\alpha j}}{\partial x_j}\,\middle|\,\Psi\right\rangle_\Delta f_{\Delta,\Phi}\left(\Psi;x_i,t\right)\right] \\ &\quad -\frac{\partial}{\partial \psi_\alpha}\left[\left\langle S_\alpha\left(\Phi\right) \mid \Psi\right\rangle_\Delta f_{\Delta,\Phi}\left(\Psi;x_i,t\right)\right]. \end{aligned} \tag{2.41}$$

The whole transport equation of ϕ_α is conditioned to its sample space Ψ inside the composition FDF transport equation.

The first right hand side of Eq. (2.41) is the unclosed convective term. We can change the ψ_α derivative by the spatial one (COLUCCI et al., 1998)

$$-\frac{\partial \langle u_j \mid \Psi \rangle_\Delta f_{\Delta,\Phi}\left(\Psi; x_i, t\right)}{\partial x_j}. \tag{2.42}$$

A similar procedure used in LES,

$$\langle u_j \phi_\alpha \rangle_\Delta = \langle u_j \rangle_\Delta \langle \phi_\alpha \rangle_\Delta + \left[\langle u_j \phi_\alpha \rangle_\Delta - \langle u_j \rangle_\Delta \langle \phi_\alpha \rangle_\Delta \right], \tag{2.43}$$

is applied to divide large from subgrid scales Eq. (2.44). In fact, Eq. (2.43) is the first moment of

$$\langle u_j \mid \Psi \rangle_\Delta f_{\Delta,\Phi} = \langle u_j \rangle_\Delta f_{\Delta,\Phi} + \left[\langle u_j \mid \Psi \rangle_\Delta - \langle u_j \rangle_\Delta \right] f_{\Delta,\Phi}, \tag{2.44}$$

which gives us a closed term, which represents the filtered convection of $f_{\Delta,\Phi}$ in Cartesian space, and an unclosed term called subgrid convective flux, which represents the effects of unresolved SGS convection. Large eddy simulation must provide $\langle u_j \rangle_\Delta$ (COLUCCI et al., 1998).

Finally, the filtered density function $f_{\Delta,\Phi}$ transport equation becomes

$$\begin{aligned} \frac{\partial f_{\Delta,\Phi}}{\partial t} + \frac{\partial \langle u_j \rangle_\Delta f_{\Delta,\Phi}}{\partial x_j} &= -\frac{\partial \left[\langle u_j \mid \Psi \rangle_\Delta - \langle u_j \rangle_\Delta \right] f_{\Delta,\Phi}}{\partial x_j} \\ &\quad + \frac{\partial}{\partial \psi_\alpha} \left[\left\langle \frac{\partial J_{\alpha j}}{\partial x_j} \middle| \Psi \right\rangle_\Delta f_{\Delta,\Phi} \right] - \frac{\partial \left[S_\alpha\left(\Phi\right) f_{\Delta,\Phi} \right]}{\partial \psi_\alpha}, \end{aligned} \tag{2.45}$$

whose unclosed terms we must model. These terms are the SGS convective flux — mentioned above — and the diffusion term, which expresses the influence of molecular diffusion on FDF transport. The last right hand side term is a closed form of the chemical reaction rate.

The SGS convective flux is modeled as

$$\left[\langle u_j \mid \Psi \rangle_\Delta - \langle u_j \rangle_\Delta \right] f_{\Delta,\Phi} = -\Gamma^{SGS} \frac{\partial f_{\Delta,\Phi}}{\partial x_j}, \tag{2.46}$$

whose first moment is given by Eq. (2.19).

We may divide the diffusion term into two parts

$$-\frac{\partial}{\partial \psi_\alpha}\left[\left\langle \frac{\partial J_{\alpha j}}{\partial x_j}\middle| \Psi\right\rangle_\Delta f_{\Delta,\Phi}\right] = \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial f_{\Delta,\Phi}}{\partial x_j}\right) - \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta}\left[\left\langle \Gamma \frac{\partial \phi_\alpha}{\partial x_j}\frac{\partial \phi_\beta}{\partial x_j}\middle| \Psi\right\rangle_\Delta f_{\Delta,\Phi}\right]. \quad (2.47)$$

The first right hand side term of Eq. (2.47) represents the effects of molecular diffusion in spatial transport of FDF, while the second — called conditional SGS diffusion — represents the dissipative nature of subgrid scalar mixing. We can model the conditional SGS diffusion using, for instance, the Interaction by Exchange with the Mean (IEM) closure (COLUCCI et al., 1998)

$$-\frac{\partial}{\partial \psi_\alpha}\left[\Omega_m\left(\psi_\alpha - \langle \phi_\alpha \rangle\right) f_{\Delta,\Phi}\right], \quad (2.48)$$

where Ω_m is the subgrid frequency of mixing. This frequency depends on large scale variables; in LES, we can relate it with the SGS diffusion coefficient, the filter width and an assumed constant mechanical-to-scalar time-scale ratio C_Ω (VEDOVOTO, 2011),

$$\Omega_m = C_\Omega\left(\Gamma + \Gamma^{SGS}\right)/\Delta^2. \quad (2.49)$$

The final form of the composition FDF transport equation is

$$\begin{aligned}\frac{\partial f_{\Delta,\Phi}}{\partial t} + \frac{\partial \langle u_j \rangle_\Delta f_{\Delta,\Phi}}{\partial x_j} &= \frac{\partial}{\partial x_j}\left(\left(\Gamma + \Gamma^{SGS}\right)\frac{\partial f_{\Delta,\Phi}}{\partial x_j}\right) \\ &+ \frac{\partial}{\partial \psi_\alpha}\left[\Omega_m\left(\psi_\alpha - \langle \phi_\alpha \rangle_\Delta\right) f_{\Delta,\Phi}\right] - \frac{\partial\left[S_\alpha(\Phi) f_{\Delta,\Phi}\right]}{\partial \psi_\alpha}, \quad (2.50)\end{aligned}$$

with all closures applied.

2.4 Stochastic Differential Equations

We used a system of stochastic differential equations equivalent to Eq. (2.50), which gives us all composition statistics. Although we can handle these equations based on Eulerian and Lagrangian frameworks, we chose the latter approach because it is more accurate than the former — due to minimum numerical diffusion (VEDOVOTO, 2011).

Both Eulerian and Lagrangian Monte Carlo use notional particles to carry the composition FDF. Those particles are transported in Cartesian space by large scales, SGS convection and molecular diffusion; particle composition changes due to SGS diffusion and chemical reaction. However, notional particles are mesh independent in Lagrangian approach, while they are mesh dependent in Eulerian (COLUCCI et al., 1998).

The following system of stochastic differential equations models the general diffusion process (COLUCCI et al., 1998):

$$dX_i\left(t\right) = D_i\left(\vec{X}\left(t\right), t\right)dt + E\left(\vec{X}\left(t\right), t\right)dW_i\left(t\right), \tag{2.51}$$

where the coefficients come from a comparison between Eq. (2.51) and Eq. (2.50)

$$E \equiv \sqrt{2\left(\Gamma + \Gamma^{SGS}\right)}, \tag{2.52}$$

$$D_i \equiv \left\langle u_i \right\rangle_\Delta + \frac{\partial\left(\Gamma + \Gamma^{SGS}\right)}{\partial x_i}, \tag{2.53}$$

which need the filtered velocity and the SGS molecular diffusivity from large eddy simulation.

The stochastic equation that describes the particles spatial evolution is

$$dX_i\left(t\right) = \left[\left\langle u_i \right\rangle_\Delta + \frac{\partial\left(\Gamma + \Gamma^{SGS}\right)}{\partial x_i}\right]dt + \left[2\left(\Gamma + \Gamma^{SGS}\right)\right]^{1/2} dW_i, \tag{2.54}$$

where $dW\left(t\right)$ is a statistical independent increment — the Wiener noise.

The stochastic equation that describes the evolution of particles composition is

$$\frac{d\phi_\alpha^*}{dt} = -\Omega_m\left(\phi_\alpha^* - \left\langle \phi_\alpha \right\rangle_\Delta\right) + S_\alpha, \tag{2.55}$$

where, ϕ_α^* is the α species composition of the notional particle and Ω_m is the subgrid frequency of mixing Eq. (2.49). The term that comprises this frequency is the IEM micromixing model. It is a current subject of research which requires more development (ORBEGOSO, 2007).

One can calculate the composition first moment $\left\langle \phi_\alpha \right\rangle_\Delta$ and its higher moments (e.g., variance) using the particles ensemble in each Eulerian cell described in chapter 3.

We numerically treated Eqs. (2.54) and (2.55) separately due to simplicity and to preserve the Markovian character of the diffusion process (VEDOVOTO, 2011)

$$X_i^p\left(t_{n+1}\right) = X_i^p\left(t_n\right) + D_i^p\left(t_n\right)\Delta t + E^p\left(t_n\right)\Delta t^{1/2}\xi_i^p\left(t_n\right), \tag{2.56}$$

where, n stands for the current time step, p represents the particle and ξ_i^p is a random variable with normalized Gaussian distribution.

CHAPTER III

LAGRANGIAN MAP

This chapter aims at exposing the main features of the Lagrangian map, which we developed during this work. It is a Lagrangian framework linked to the well established Finite Differences framework of the in-house built AMR3D code (VILLAR, 2007; LIMA, 2012). The map consequently supports a hybrid LES / Lagrangian FDF method we have implemented and applied.

3.1 Adaptive, block-structured mesh

The adaptive, block-structured mesh — hereafter referred by the acronym of the Adaptive Mesh Refinement (AMR) family methods — is a computational technique used to dynamically achieve high numerical precision on flow regions of interest using Finite Differences (FD) method or analogous. It greatly contributes to minimize the required computational resource and the simulation time because distributes cell nodes heterogeneously throughout flow domain. Figure (3.1) shows three levels of mesh resolution, which are more refined near the top and right edges.

The greatest disadvantage of an adaptive, block-structured mesh is its inflexibility to fit complex domains and immersed objects. However, Lagrangian based methods like Immersed Boundary and Front Tracking have been successfully applied to overcome that limitation (VILLAR, 2007). The particles can stand spatially fixed or move according with

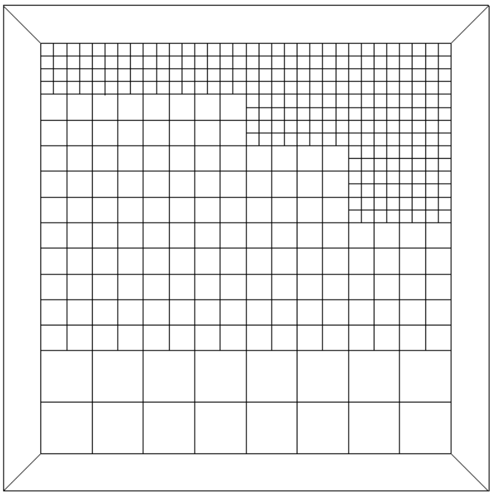


Figure 3.1: Longitudinal cross-section of AMR in 3D domain with 3 grid levels.

domain borders and objects displacements. In both cases, the link between mesh and particle variables occurs by interpolation.

In the AMR3D code, the flow variables (e.g., velocity and species mass fraction) evolve on adaptive, block-structured mesh. The refinement criterion is based on the magnitude of the vorticity field. If we set a mesh with 3 grid levels and a refinement range of 5%, the first and second finest mesh will cover the regions of the flow within $[0.95 - 1.0]$ and $[0.9 - 0.95]$ of the greatest vorticity intensity; the coarsest mesh will cover all flow domain.

That mesh is easily parallelized compared to unstructured mesh (LIMA, 2012). Lima (2012) discoursed on AMR3D parallelization methods, and used Message Passing Interface (MPI) to communicate Eulerian based data among processes. We used the same approach to communicate Lagrangian information using the parallel Lagrangian map, as it will be seen in section 3.3.

3.2 Eulerian / Lagrangian link

Many are the flows which involve Lagrangian elements, like particles for instance. The computational particle can represent distinctive natures, for example a particulate in gas-solid flow or a stochastic element in a reactive flow, with physical and numerical essence, respectively. Nowadays, the majority of literature addresses Lagrangian transport on unstructured mesh (ANSARI et al., 2011); few papers relate this transport on AMR (DUBEY;

ANTYPAS; DALEY, 2011).

The computational model of a system — preceded by the physical, the mathematical and the numerical models — is crucial considering the quality of a Computational Fluid Dynamics (CFD) software. The Eulerian / Lagrangian modeling requires a data structure able to exchange information between these two frameworks. Moreover, the direct access to the ensemble of particles contained in each cell is essential to the Lagrangian FDF code performance, especially regarding algorithms dependent on sorting (FOX, 2003).

Two questions arise when we observe the transport of particles in Eulerian domain — specially on AMR. First, which particles dwell in a given cell; second, which is the host cell of a specific particle. The Lagrangian map answers both questions. It is composed by a multi-level hash table that provides direct access to the ensemble of particles contained in each cell Fig. (3.2), and uses a per level search tool based on algebraic equation Eq. (3.1) to find the cell in which the particle resides.

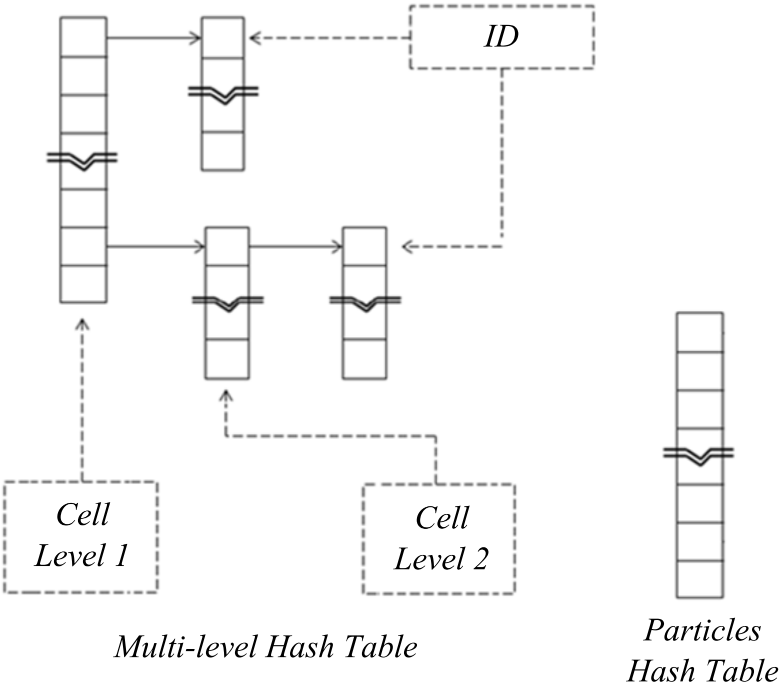


Figure 3.2: Lagrangian map: computational model composed by a multi-level hash table of identifiers (ID) and a particles hash table.

Figure (3.2) shows that the first hash tables are always cell-type and the last one is always ID-type. The cells in which particles dwell are called *visible cells*. Since a multi-level hash table depends on Eulerian mesh, the code must rebuild its structure at each mesh refinement. In a 3D domain, each coarse cell originates eight fine cells.

3.2.1 Hash table

Data structures store data into and acquire data from random-access memory and hard disc. Many are based on predefined size arrays, which use an integer index to address data, others are based on linked lists, which dynamically connect data using pointers; hash tables provide a mixture of these features. Shaffer (2013) described several data structures and their applicability.

Hash table is a dynamic data structure that allows direct reference to storage and access data. The reference occurs by a key, which can be an integer, a character or an user-defined type. Therefore, each data of a collection must have a unique key. The pairing rule used to attend this request is called *hash function*. When we try to storage two data using the same key, a *collision* occurs.

Figure (3.3) shows the AMR is an inherent collision system: the key of each cell is its index (i, j, k), which is the same to a refined cell and the correspondent low-level coarse cell. It occurs because these cells have the same origin.

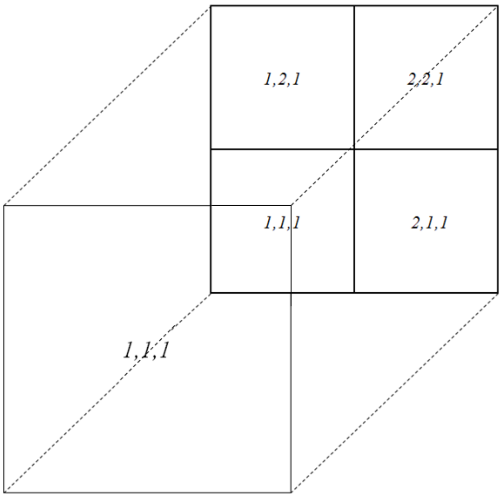


Figure 3.3: The same key $(1, 1, 1)$ used to the coarse and refined cells leads to hash table collision.

A *collision resolution policy* is practiced by applying the *open hashing* or the *closed hashing* techniques. The policy is called closed hashing when it saves the conflicting data into the same hash table, just using another key. When it uses another hash table to save the data, it is called open hashing (SHAFFER, 2013).

The developed Lagrangian framework applies the open hashing using the multi-level hash table based on *Uthash* (HANSON, 2013). Uthash is a C header file developed under the revised BSD license in 2006 by Troy D. Hanson due to the lack of this kind of data structure in the standard C library. Several commercial and academic software have incorporated it since the first release. Uthash provides tools to create and manipulate hash tables (e.g., addition, deletion and searching), which are $O(1)$ for time processing.

3.2.2 Search algorithm

Many computational applications foment business, help communication and model the physical world (e.g., service, engineering and scientific software). They all need some kind of search mechanism to operate over a database. The code can invoke them several times — what makes these algorithms attractive in terms of efficiency improvement.

Search algorithms are classified based on the desired response. The *exact-match* and the *range query* verify a data existence and the relative position of a data among a collection, respectively. The search approach also varies according to the kind of information — whether data is sorted or not, small or large, integer or floating-point, directly or indirectly accessed (SHAFFER, 2013).

Sequential, binary and dictionary based searches apply to orderly collections of data. The sequential search presents $O(n)$ — n stands for the amount of data — and analyses each value, comparing them with the sought data. It is inefficient for great collections and frequently demanded operations. The binary search is $O(\log(n))$ and split the investigation in consecutively two sub domains. The dictionary based search is similar to the binary one, though more efficient. It is $O(\log\log(n))$ because it knows how data are distributed, optimizing the domain split (SHAFFER, 2013). Since mesh width is uniform per level in AMR, the search tool can use a direct mechanism based on the algebraic equation

$$index = f\left(\frac{\vec{P}-\vec{P}_0}{\Delta^l}\right), \tag{3.1}$$

where, $\vec{P}$ is the particle position, $\vec{P}_0$ is the domain origin and Δ^l is the mesh width at current searched level, l. This algorithm is $O(1)$ per level as shown in Fig. (3.4).

3.2.3 Asymptotic Algorithm Analysis

The asymptotic algorithm analysis aims to discover the computational cost growth rate of an algorithm (e.g., simulation time *vs.* number of processed data). We can use simplified techniques in order to understand that pattern. Upper bound, the big-Oh notation, and lower bound, the big-Omega notation, represent the maximum and minimum algorithm computational cost, respectively.

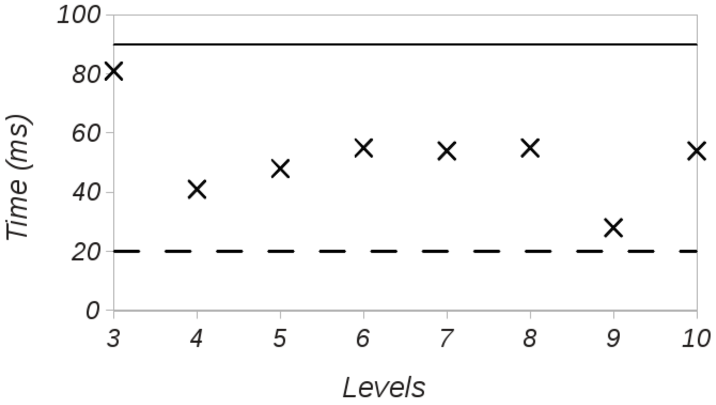


Figure 3.4: Time, in milliseconds, for a random particle search using the Lagrangian map. The continuous and dashed line stand for the upper $(O(1))$ and lower bounds $(\Omega(1))$, respectively.

The upper and lower bounds are defined as follow (SHAFFER, 2013):

- For $T(n)$ a non-negatively valued function, $T(n)$ is $O(f(n))$ if there exists two positive constants c and n_0 such that $T(n) \leq cf(n)$ for all $n > n_0$.
- For $T(n)$ a non-negatively valued function, $T(n)$ is $\Omega(g(n))$ if there exists two positive constants c and n_0 such that $T(n) \geq cf(n)$ for all $n > n_0$.

In this master thesis, we evaluated the computational cost of the developed Lagrangian map, which handles the transport of particles on AMR. We present results of the Lagrangian map running a Intel® Core™ i5-3450, 3.10 GHz CPU, 1,600.00 MHz speed computer.

Figure (3.4) shows the search time for locating a particle on AMR using the Lagrangian map search tool. The horizontal axis represents the number of AMR levels — which corresponds to meshes from 512 up to one billion cells.

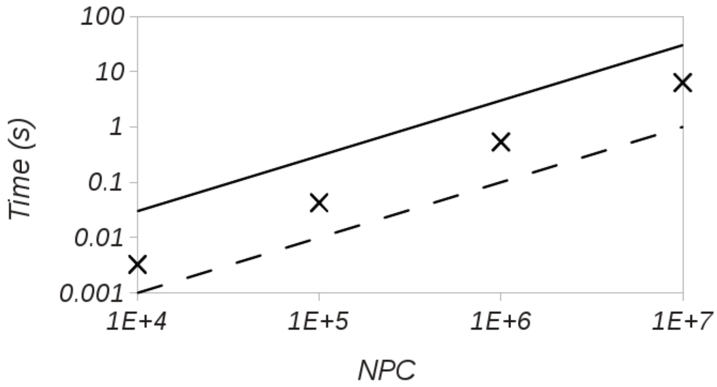


Figure 3.5: Time, in seconds, for particles transport between two cells using the Lagrangian Map; *NPC* stands for *number of particles per cell.* The continuous and dashed line stand for the upper ($O(n)$) and lower bounds ($\Omega(n)$), respectively.

During Lagrangian transport, the algorithm iterates the hash table of particles while adds each one in a neighbor cell and removes it from the original cell. Figure (3.5) shows the transport of up to ten million particles. The computational cost is linear because of hash table loop.

3.3 Lagrangian distributed processing

The parallel Lagrangian map is an extension of the Lagrangian map that handles the transport of particles on AMR through computer processes based on distributed processing. It is composed by a multi-level hash table, which stores particles identifier, and by an independent hash table that saves particles information using these IDs Fig. (3.2).

Since CFD simulations may require millions of cells and billions of particles, distributed processing is indispensable. In order to achieve high efficiency in these applications, the load of each process must be equalized and the communication among processes must be minimized. Load balancing optimization is required once in static, block-structured mesh simulations; however, the simulation may perform this task several times when the mesh is adaptive.

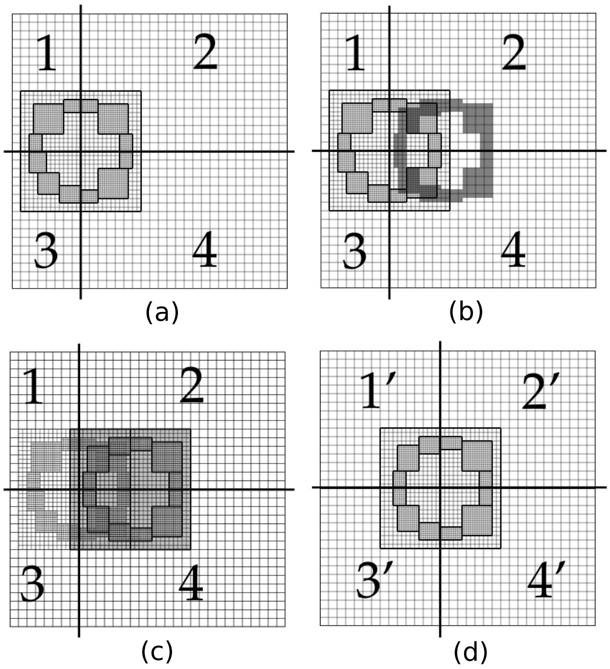


Figure 3.6: AMR bisection during refinement — 4 processes: (a) initial mesh, (b) select new refined cells, (c) refine the mesh, (d) new domain partition (LIMA, 2012).

In the AMR3D code, the Recursive Coordinate Bisection method is used to partition the computational domain. This method is low cost, results in satisfactory load balancing and works with rectilinear spaces (LIMA, 2012). The RCB divides sequentially the grid into two parts in order to guarantee an equal load in both sides. Partition occurs perpendicularly to each direction until the split sub domains equal to the number of available processes. The Zoltan is a library which assists domain partition before use the Recursive Coordinate Bisection method. It has an ensemble of load balancing tools and parallel data management — among them the weight assignment to each visible cell (LIMA, 2012).

Figure 3.6 shows a mesh bisection during refinement. Mesh refinement can occur many times until load balancing is invoked. Although optimized Eulerian and Lagrangian load balancing have similar trends (DUBEY; ANTYPAS; DALEY, 2011), they are not the same. Since simulation obey Eulerian partition, it does not reach optimum Lagrangian load balancing.

3.3.1 Message Passing Interface

The Message Passing Interface (MPI) is a set of standards used to communicate any kind of data among computer processes. Although it is a recent information technology, launched about 20 years ago, nowadays it is widely applied in the prevailing distributed-processing software.

Three elements are ever present during parallel transport of data: the *communicator* (group of processes), the *rank* (identifier of each process) and the *tag* (identifier of the transferred data). Every distributed processing application running MPI must start and finalize with *MPI_Init* and *MPI_Finalize* function calls, respectively; wherein the communicator is created and destroyed with the entire MPI environment (KENDALL, 2013).

In the AMR3D code, both AMR structure and the Finite Differences framework are developed in Fortran. Therefore, Fortran functions initialize the communicator and further transfer it to C environment, where Lagrangian FDF module resides.

The point-to-point communication occurs when only two processes communicate each other; otherwise, it is called collective communication. Collective communication includes every processes and demands all of them be synchronized. The *MPI_Barrier* function performs this task.

We applied collective communication in parallel transport of particles to inform all processes the number of particles each one must send and receive from each neighbor process. On the other hand, we used point-to-point communication to send and receive particles' buffer among these processes — in a manner to avoid *deadlock*. Deadlock occurs when multiple processes stop their tasks waiting for a response from another process, or when the message exchange among them has a chain form (KENDALL, 2013).

Other useful MPI functions are *MPI_Bcast*, *MPI_Scatter*, *MPI_Gather*, *MPI_Allgather*, *MPI_Reduce* and *MPI_Allreduce*.

The *MPI_Bcast* is an elementary collective communication function and is associated with the broadcasting of a unique message from the root process to all processes. This function consists of sending and receiving point-to-point procedures. In order to maximize the network, each process also communicates the data after had received it. Other broadcasting function is *MPI_Scatter*; it can send different size of data to each process (KENDALL, 2013).

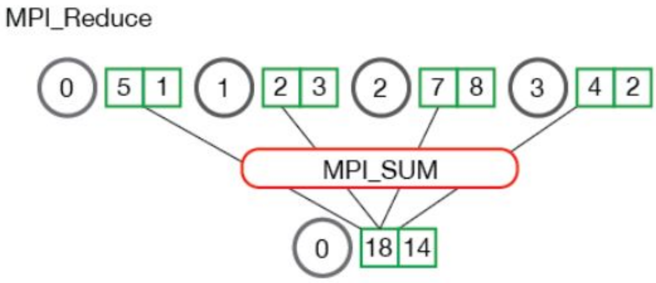


Figure 3.7: *MPI_Sum* operation upon arrays from 4 processes — process 0 is the root. (KENDALL, 2013).

The *MPI_Gather* groups in the root process the messages of every neighbors and stores them into an array of type *MPI_Datatype*. We must invoke *MPI_Allgather* when all processes need to know these grouped messages.

Figure (3.7) shows one of the available mathematical operations to be performed upon transferred data using *MPI_Reduce* function. Some operations *MPI_Op* are *MPI_Max*, *MPI_Min*, *MPI_Sum* and *MPI_Prod*. When the *MPI_Reduce* receives a single scalar from each process, the output is also a scalar quantity; otherwise, the result is an equal size array Fig. (3.7). Many distributed-processing algorithms need some result shared throughout the communicator after had manipulated a data. In this case, we must use the *MPI_Allreduce* function (KENDALL, 2013).

3.3.2 ID controller

Just like Eulerian mesh, particles also follow domain partition rule i.e., their positions determine to which process they belong. Since each particle has a unique identifier (ID), which is known just in the local process, all processes must have an independent ID controller.

Moreover, the flow dictates how many particles must be added to and deleted from a local Lagrangian map. It demands a dynamic computational tool to manage each of these ID collections. We adopted a *stack* to perform this task. Stack is a computational model which exchanges data just by one of its sides. The statement *"last in, first out"* rules it (SHAFFER, 2013).

Figure (3.8) shows all stacks initialize with a set of integer numbers; while the simulation adds particles in flow domain at initial stage, stacks assign an ID to each added particle.

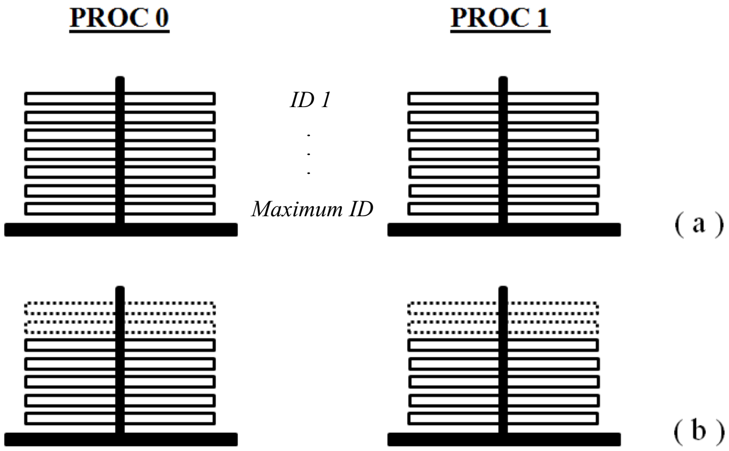


Figure 3.8: (a) Stacks initialize with *Maximum ID*. (b) Stacks assign two IDs to added particles (dashed line).

When a particle leaves local domain, its identifier returns to the stack of origin. If this particle enters into a neighbor process, the local stack provides a new ID to that particle. This mechanism avoids hash table collision.

Figure (3.9) shows a particle transfer between two processes. Firstly, both stacks start with IDs 1, 2, 3 and 4. Further, two particles enter into each domain, and receive the available IDs from the local stacks Fig. (3.9-a). Finally, the particle whose ID is *number 2* moves from *PROC 0* to *PROC 1*. At this stage, the local stack provides it a new ID (*number 3*), while the stack of origin receives the older ID (*number 2*) Fig. (3.9-b).

We can use other approaches to handle particles IDs in distributed processing environment. For example, it is possible to use a *counter* in each process, which resets itself when the number of used IDs achieves the maximum *int* or *long int* size. We chose the stack model because it is simple to develop and understand — if we expect one thousand particles in global flow domain at initial stage, and the simulation run using four processes, it is more natural to have particles IDs ranging from 1 to 2500 rather than IDs ranging from 1 to $2,147,483,647$, for example.

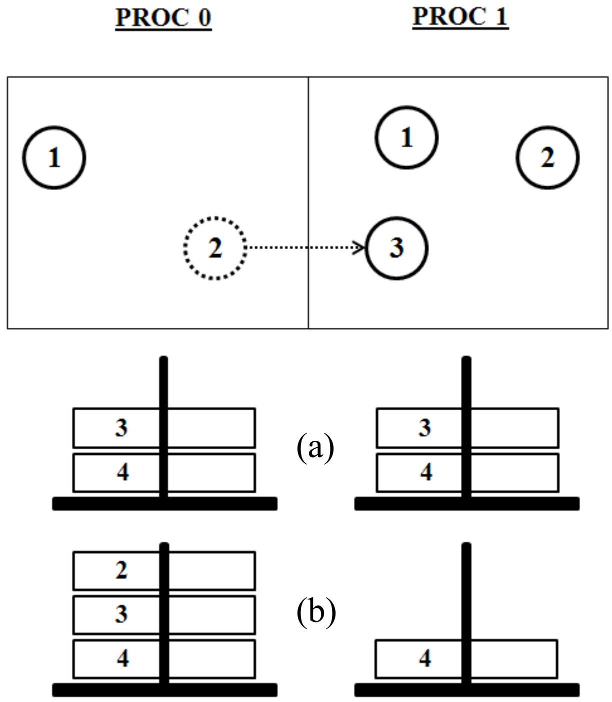


Figure 3.9: ID controller. (a) Before and (b) after particle transfer.

3.3.3 Particles initialization

The Lagrangian map structure is similar to a computational *tree* — visible cells in block-structured mesh are the *tree leaves* Fig. (3.10). Visible cells distinguish from others because just the last layer of cells can hold particles.

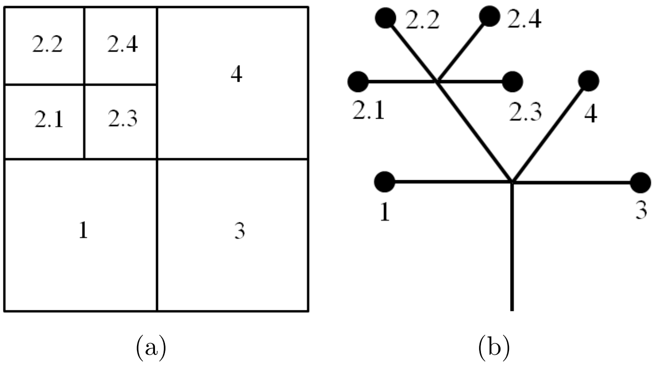
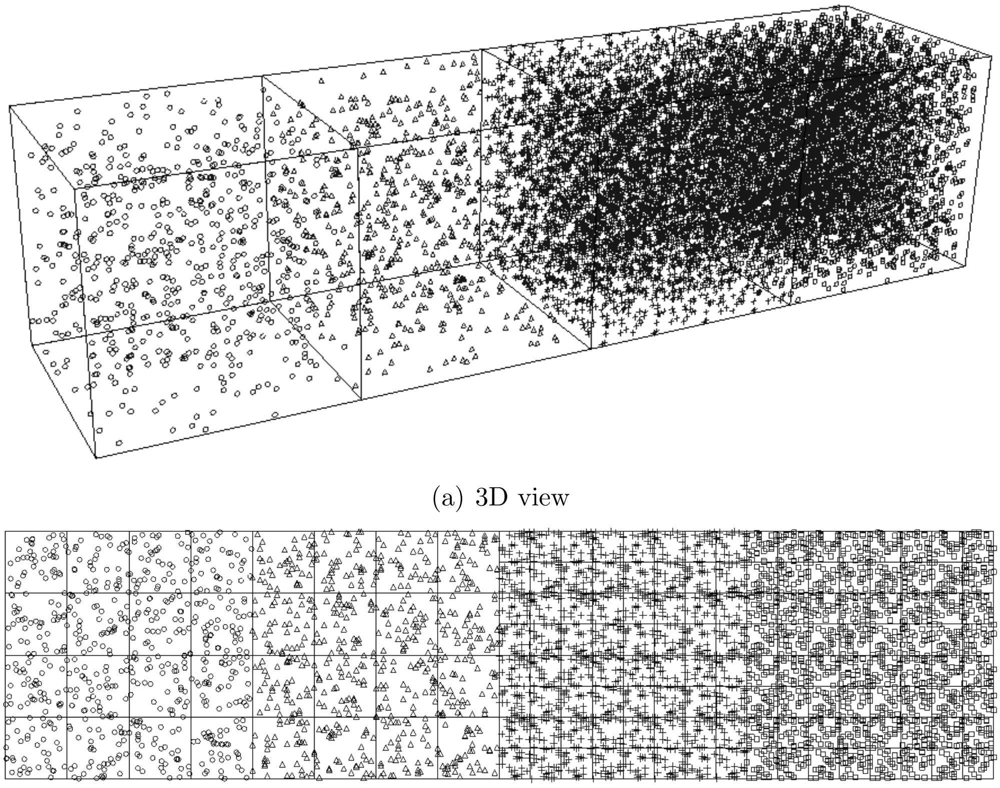


Figure 3.10: (a) Two dimensions block-structured mesh. (b) Lagrangian map structure — the black discs represent the visible cells (leaves).

This section illustrates how particles initialize using two configurations of 3D domains. In the first case, a rectangular longitudinal-section domain with 4 m length and unit transversal-area is partitioned into 4 processes. The processes are aligned in one direction and have equal size. The first and second processes have each one 4^3 visible cells, whilst the third and forth have each one 8^3 Fig. (3.11).



(a) 3D view

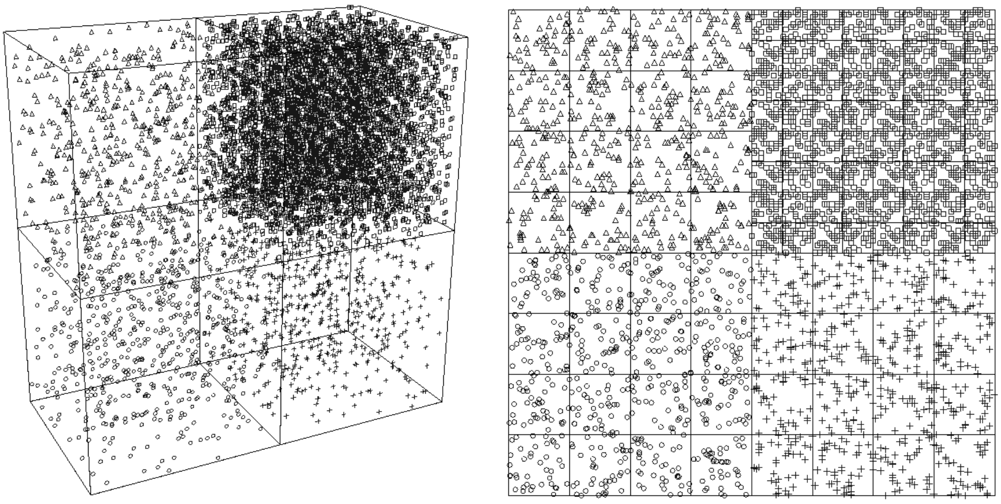
(b) Longitudinal-section

Figure 3.11: First case — from first to forth processes, particles are illustrated by circles, triangles, crosses and squares, respectively. The longitudinal-section of the mesh shows all particles in the aligned cells.

In the second case, a $4\,m^2$ longitudinal-section, unit depth domain is partitioned into 4 processes. The processes are aligned in two directions and have equal size. The first three processes have each one 4^3 visible cells, whilst the forth has 8^3 Fig. (3.12).

Figures (3.11) and (3.12) show how particles initialize per cell — each cell has $n = 10$ particles uniformly distributed. The largest particle density is found where mesh is more refined.

We used a recursive function to iterate the visible cells stored in Lagrangian map. Once the algorithm finds these cells, it adds particle IDs using the stack (section 3.3.2). Each cell receives n identifiers, independently of its size and location in flow domain Figs. (3.11) and (3.12). This approach is appropriate to Lagrangian FDF methods, because it guarantees a



(a) 3D view

(b) Longitudinal-section

Figure 3.12: Second case — from first to forth processes, particles are illustrated by circles, triangles, crosses and squares, respectively. The longitudinal-section of the mesh shows all particles in the aligned cells.

homogeneous statistical error throughout domain (HAWORTH, 2010).

Particles can also initialize uniformly over global domain. Sections (3.3.4) covers this subject.

3.3.4 Forced transport of particles

The parallel transport of particles uses point-to-point and collective communication concepts of MPI. Initially, each process creates an *identifier / target process* list. Then, the root process gathers all lists and distributes them to the other processes. Finally, processes send and receive their particle's buffer, containing information such as position, weight and composition value.

For both domain configurations described in section (3.3.3), $n = 25$ particles were uniformly added in each process. In the first case Fig. (3.13), $\vec{V} = (10\vec{i})$ m/s was imposed in longitudinal direction, whilst in the second case Fig. (3.14), $\vec{V} = (10\vec{i} + 10\vec{j})$ m/s was imposed in longitudinal and transversal directions. Both simulations used the statistical independent increment of Eq. (2.54). It provides a random walk additionally to the advection linear translation. All boundary conditions are wall-type, which reflect the particle with $\pi/2$ *rad* from the incident angle.

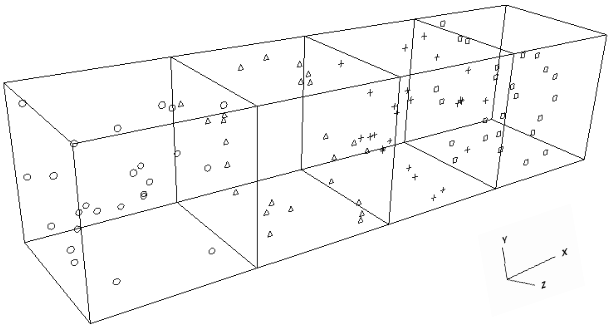


Figure 3.13: 3D view of particles uniformly distributed in domain (first case). From first to forth processes, particles are illustrated by circles, triangles, crosses and squares, respectively.

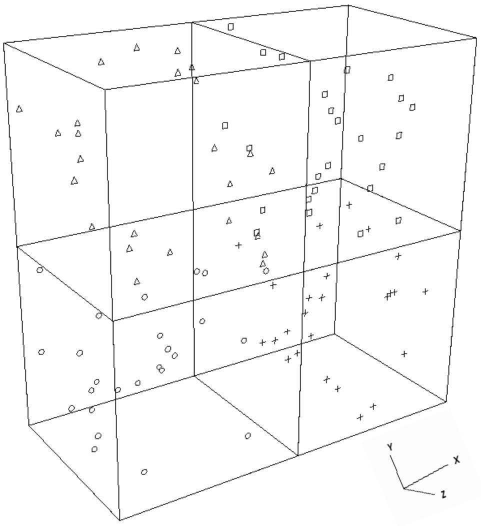


Figure 3.14: 3D view of particles uniformly distributed in domain (second case). From first to forth processes, particles are illustrated by circles, triangles, crosses and squares, respectively.

Figure (3.15) shows five simulation time steps for the first simulation case. As particles move, they accumulate near the last process wall. In this configuration, each process transfers information to just one neighbor.

Figure (3.16) shows four simulation time steps for the second simulation case. As particles move, they accumulate near the intersection of the right and superior edges in the last process. In this configuration, a process can simultaneously transfer information to more than one neighbor. The algorithm performs this task avoiding deadlock by using non-blocking send routines and blocking receive routines (KENDALL, 2013).

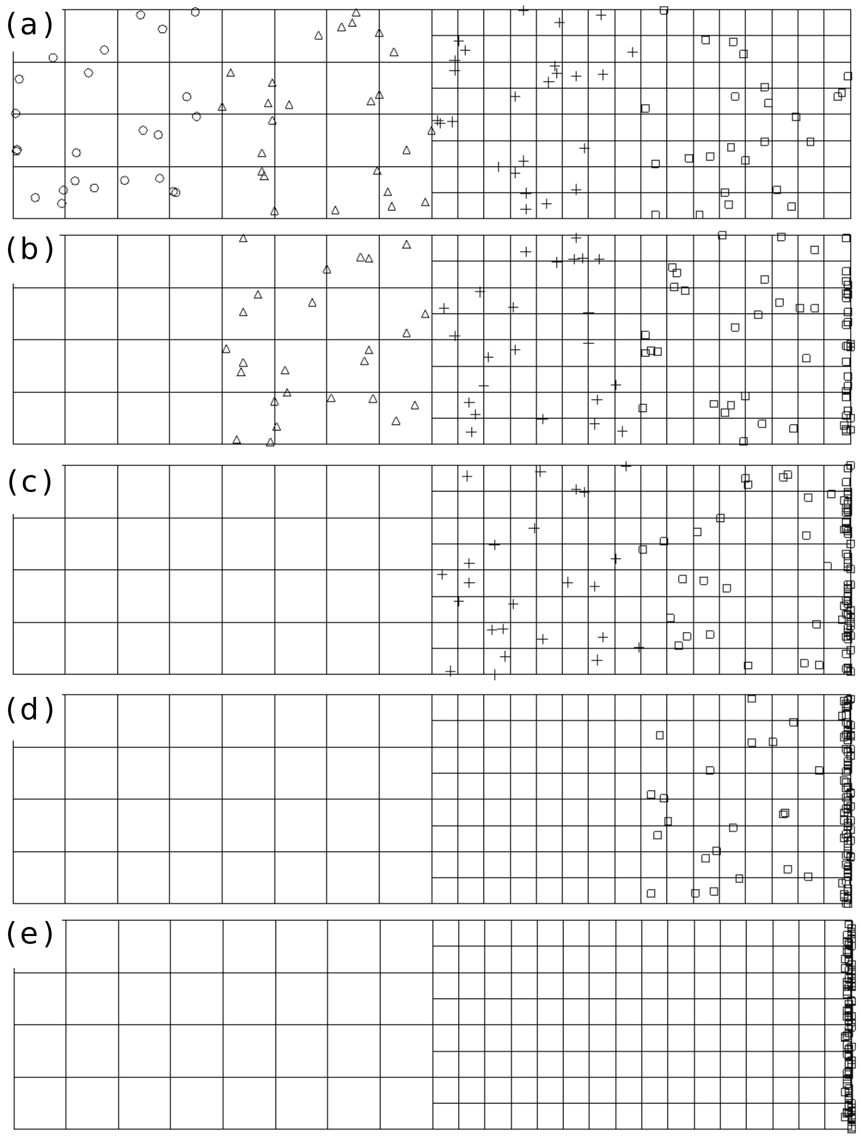


Figure 3.15: Five time steps in forced transport of particles (first case) — domain longitudinal cross-section.

3.3.5 Particles advection

The stochastic particles evolve in Cartesian space according to Eq. (2.54), in which the first right hand side term comes from the Eulerian mesh. The interpolation scheme is based on weighting distances from each cell node to particle position. The Finite Differences framework of AMR3D code (VILLAR, 2007) calculates the filtered velocity field at cell faces; further, the code interpolates these values to cell nodes. Figure (3.17) shows the same approach presented by Vedovoto (2011) using Finite Volume method.

In order to evaluate the Eulerian / Lagrangian interpolation, we simulated the starting

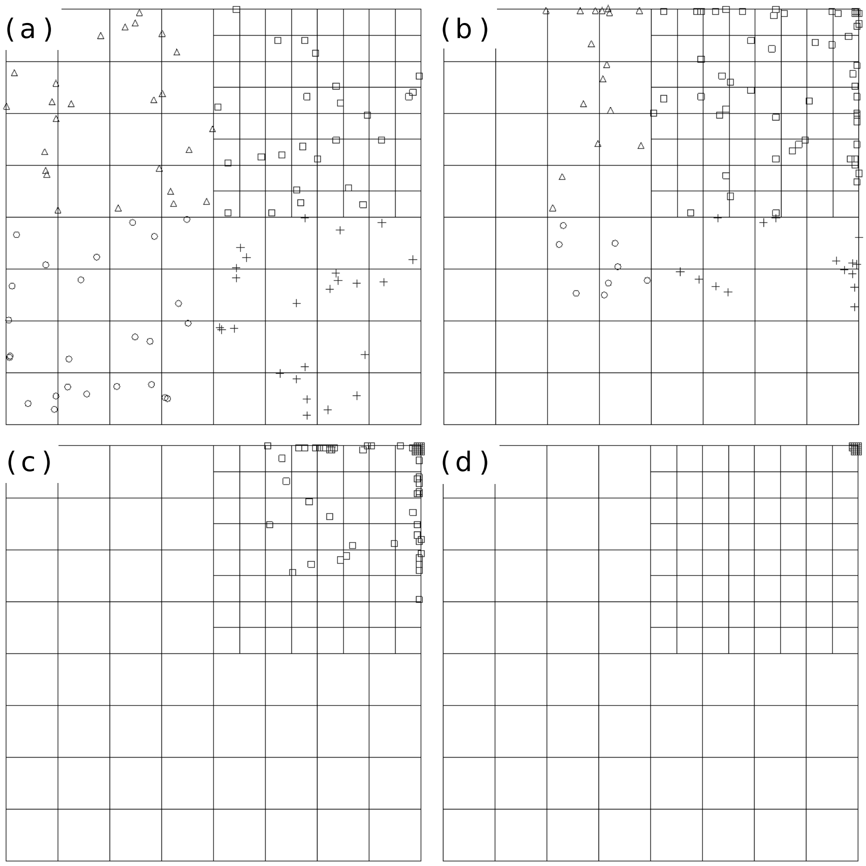


Figure 3.16: Four time steps in forced transport of particles (second case) — domain longitudinal cross-section.

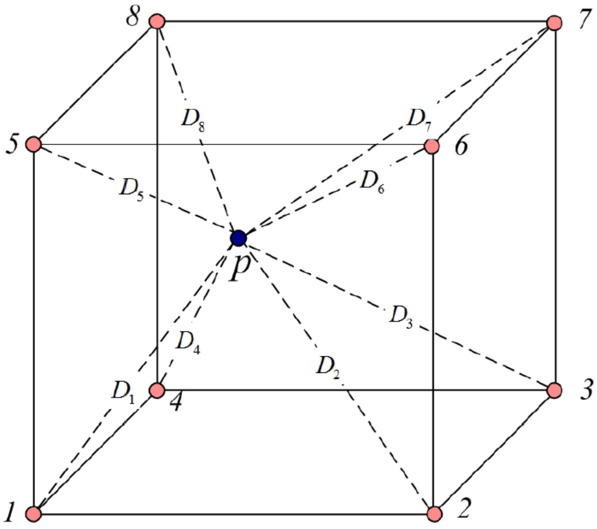


Figure 3.17: Interpolation scheme. Solid lines represent the cell edges; dashed lines stand for the distances D from each cell node to particle P (VEDOVOTO, 2011).

flow in a cubic, unit volume, lid-driven cavity. The box was filled with particles, which followed flow streamlines. The AMR is initially more refined near the lid; during simulation the mesh refinement follows flow vortices Fig. (3.20). We considered only the filtered velocity term of Eq. (2.54) in this analysis. The terms which do not depend on velocity — SGS convective flux and molecular diffusion — are treated in chapter 4.

At first time step, each visible cell receives an equal number of particles. Therefore, the region near the lid-driven has the greatest particle density. Figure (3.18) shows the initial stage of the parallel, lid-driven cavity simulation.

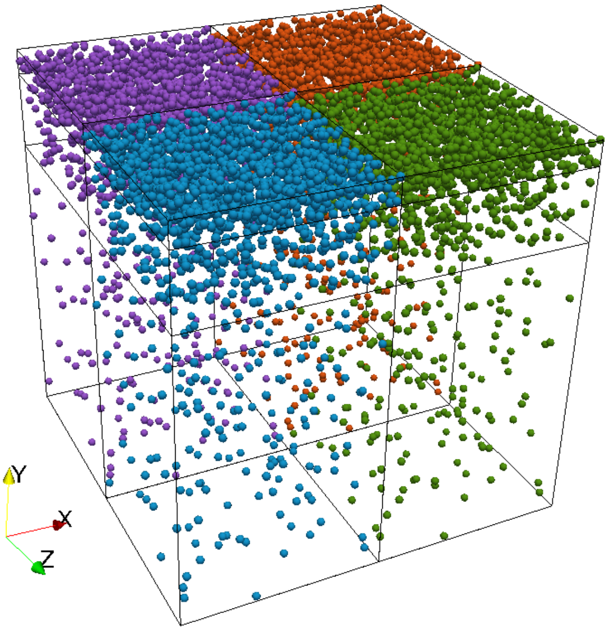
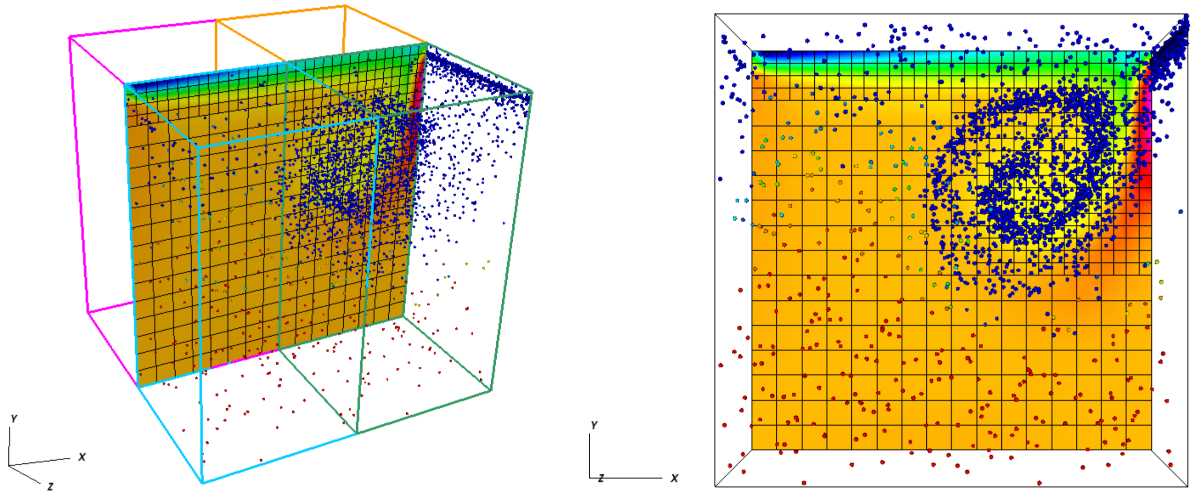


Figure 3.18: Parallel, lid-driven cavity simulation — initial condition. The host processes define particles color: blue, purple, green and red.

The upper wall moves with constant positive velocity in x direction. The flow develops inside the cavity due to velocity gradient and fluid dynamic viscosity μ. Figure (3.19) shows the longitudinal cross-section of vorticity field at $z = 0.5$. The red and green colors stand for the positive and negative flow rotation in z direction, respectively. The particles agglomerate in the vortex near the right wall.

Figure (3.19) also reveals the passive particles tend to accumulate near zero velocity regions (e.g., upper wall leading edge) and far from the vortex center. Since the Monte Carlo solver accuracy depends on the number of notional particles per cell, the simulation must keep this number under control. It explains why cloning and annihilation algorithms are essential when running the Lagrangian composition FDF code (chapter 4).

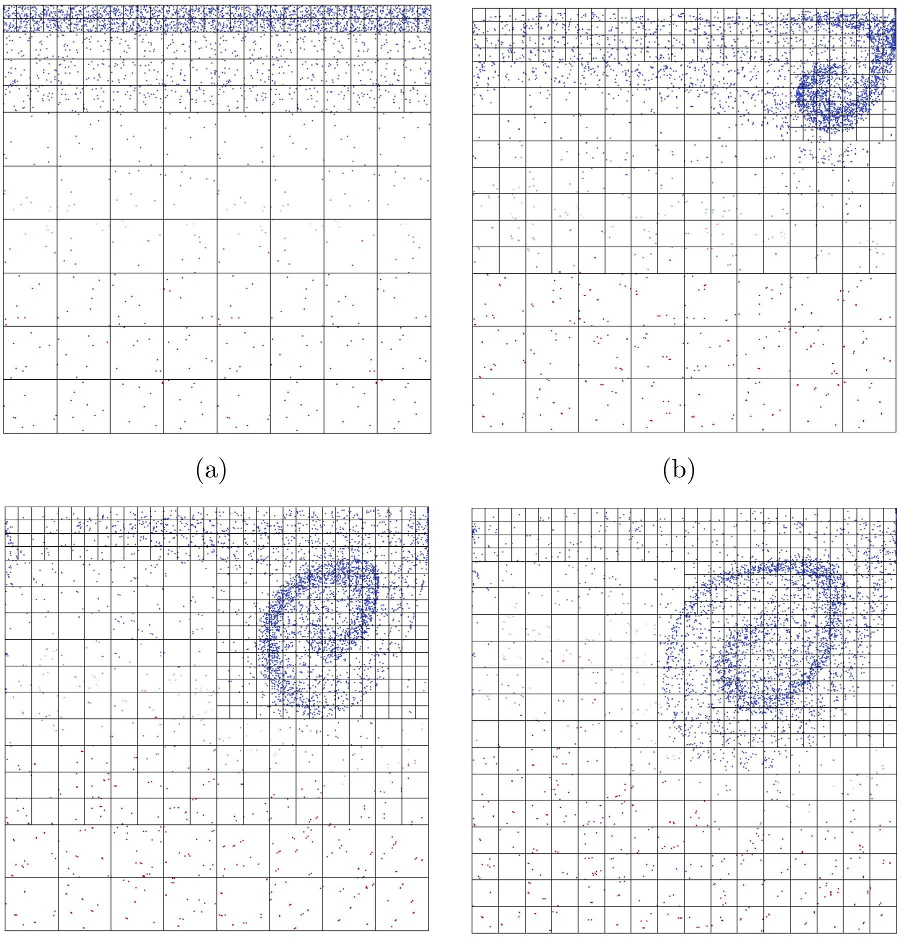


(a) 3D view

(b) Front view

Figure 3.19: Longitudinal cross-section of vorticity field at $z = 0.5$. Particles in parallel, lid-driven cavity simulation.

Moreover, the Lagrangian map must rebuild its structure at each AMR remeshing Fig. (3.20).



(a)

(b)

(c)

(d)

Figure 3.20: Longitudinal cross-section of the parallel, lid-driven cavity simulation on AMR.

CHAPTER IV

LAGRANGIAN COMPOSITION FDF CODE

This chapter describes the Lagrangian Monte Carlo algorithm embedded in the Lagrangian composition FDF code (hereafter referred by FDF code), as well as the intrinsic procedures of particles population control i.e., cloning and annihilation. Section 4.1 refers to the AMR3D code (VILLAR, 2007; LIMA, 2012) and the numerical methods used to solve the filtered momentum equations. Since FDF methods treat each Eulerian cell as a Partially Stirred Reactor (PaSR), we present an overview on this subject and also a single cell PaSR simulation. The last section covers the statistical equivalence between Eulerian (PDE solver) and Lagrangian (SDE solver) approaches, and shows mixing flow simulations on fixed and adaptive, block-structured meshes. The PDE solver deals with Eq. (2.14) and the SDE solver deals with Eqs. (2.54) and (2.55). We simulated both using the AMR3D code; we also applied the developed FDF code for SDE cases.

4.1 AMR3D code

The AMR3D code is a computational framework able to solve monophasic and multiphasic turbulent flows using Lagrangian and Eulerian based numerical methods on adaptive, block-structured mesh. We incorporated the Lagrangian composition FDF code into the AMR3D framework in order to expand its applicability to turbulent mixing flows, aiming at future developing of chemical reaction and combustion models.

Section 4.1.1 covers the temporal and spatial discretization schemes, while section 4.1.2 shows the fractional step method that couples filtered pressure with filtered momentum, guarantying mass conservation.

4.1.1 Temporal and spatial discretization

We used the Semi Backward Difference Formula (SBDF) method to discretize the filtered momentum transport equations respected to time. It treats the diffusive term implicitly and the advective term explicitly. The SBDF is efficient at low Reynolds number and large time step using the Finite Differences method (VILLAR, 2007). All simulations we performed in this master thesis fit this classification.

Integrating Eq. (2.13) using the temporal second order SBDF, results

$$\begin{aligned}\frac{\langle\rho\rangle_\Delta}{\Delta t}\left(\alpha_2\widehat{u}_i^{n+1}+\alpha_1\widehat{u}_i^{n}+\alpha_0\widehat{u}_i^{n-1}\right) &= \lambda\frac{\partial^2\widehat{u}_i^{n+1}}{\partial x_j^2}+\beta_1\cdot h_i\left(\widehat{u}_i^{n},\mu_e^{n}\right)\\ &\quad+\beta_0\cdot h_i\left(\widehat{u}_i^{n-1},\mu_e^{n-1}\right)-\frac{\partial\langle p\rangle_\Delta}{\partial x_i},\end{aligned}\tag{4.1}$$

where,

$$h_i=-\lambda\frac{\partial^2\widehat{u}_i}{\partial x_j^2}+\frac{\partial\left\langle\tau_{ij}\right\rangle_\Delta^M}{\partial x_j}-\widehat{u}_j\frac{\partial\widehat{u}_i}{\partial x_j}+\widehat{S}_i^u,\tag{4.2}$$

$$\lambda=C_\lambda\|\mu_e\|_\infty,\tag{4.3}$$

and

$$\mu_e=\mu+\mu^{SGS}.\tag{4.4}$$

We adopted $C_\lambda=2$, as used by Villar (2007). Since the time step Δt varies during simulation, α and β coefficients are

$$\alpha_0=\frac{\Delta t^2}{\Delta t_0\Delta t_1},\tag{4.5}$$

$$\alpha_1=-\frac{\Delta t_1}{\Delta t_0},\tag{4.6}$$

$$\alpha_2 = \frac{\Delta t_0 + 2\Delta t}{\Delta t_1}, \tag{4.7}$$

$$\beta_0 = -\frac{\Delta t}{\Delta t_0}, \tag{4.8}$$

and

$$\beta_1 = -\frac{\Delta t_1}{\Delta t_0}, \tag{4.9}$$

where,

$$\Delta t = t^{n+1} - t^n, \tag{4.10}$$

$$\Delta t_0 = t^n - t^{n-1}, \tag{4.11}$$

and

$$\Delta t_1 = \Delta t_0 + \Delta t. \tag{4.12}$$

Additionally, the algorithm uses first order Euler method at the first time step, instead of SBDF one (VILLAR, 2007).

We applied the Finite Differences method to spatially discretize the transport equations, and the second order Central Difference Scheme (FERZIGER; PERIC, 1996) to approximate derivatives. Scalar variables (e.g., pressure, density and viscosity) locate at cell center, and velocity components locate on cell faces according with the staggered arrangement.

4.1.2 Pressure-velocity coupling

The fractional step method solves filter pressure and velocity separated and sequentially (VILLAR, 2007). Firstly, it determines an auxiliary filtered velocity $\widehat{U_i}$ field based on the filtered momentum transport equations, but ignores mass conservation,

$$\begin{aligned}\frac{\langle\rho\rangle_\Delta}{\Delta t}\left(\alpha_2\widehat{U}_i^{n+1}+\alpha_1\widehat{u}_i^n+\alpha_0\widehat{u}_i^{n-1}\right) &= \lambda\frac{\partial^2\widehat{U}_i^{n+1}}{\partial x_j^2}+\beta_1\cdot h_i\left(\widehat{u}_i^n,\mu_e^n\right)\\ &\quad+\beta_0\cdot h_i\left(\widehat{u}_i^{n-1},\mu_e^{n-1}\right)-\frac{\partial\langle p\rangle_\Delta}{\partial x_j}.\end{aligned} \tag{4.13}$$

Further, the Poisson equation provides an estimated filtered pressure $\langle q\rangle_\Delta$ based on the auxiliary filtered velocity field

$$\frac{\partial}{\partial x_j}\left(\frac{1}{\langle\rho\rangle_\Delta}\frac{\partial\left\langle q\right\rangle_\Delta^{n+1}}{\partial x_j}\right)=\frac{\alpha_2}{\Delta t}\frac{\partial\widehat{U}_i^{n+1}}{\partial x_j}. \tag{4.14}$$

The method forces the auxiliary filtered velocity field to fit mass conservation, correcting $\widehat{u}_i$ with the estimated filtered pressure

$$\widehat{u}_i^{n+1}=\widehat{U}_i^{n+1}-\frac{\Delta t}{\alpha_2\langle\rho\rangle_\Delta}\frac{\partial{\langle q\rangle_\Delta}^{n+1}}{\partial x_j}. \tag{4.15}$$

After these steps, the filtered velocity field obeys mass conservation. We obtain the corrected filtered pressure field using the estimated one

$$\langle p\rangle_\Delta^{n+1}=\langle p\rangle_\Delta^n+\langle q\rangle_\Delta^{n+1}. \tag{4.16}$$

The AMR3D code uses the Multigrid technique, which employs the successive over-relaxation iterative solver to provide numerical solution for the Poisson equation and the diffusive term of momentum equation — which is treated implicitly by the temporal discretization scheme. It is considered one of the fastest and efficient algorithm to solve linear systems, and has a grid independent convergence rate (VILLAR, 2007).

4.2 Lagrangian Monte Carlo algorithm

The FDF code works with the Large Eddy Simulation module. Figure (4.1) shows the flowchart of the FDF code.

At first time step, the algorithm initializes particles in the flow domain obeying the initial condition for the composition field. Inside the time loop, the code interpolates from

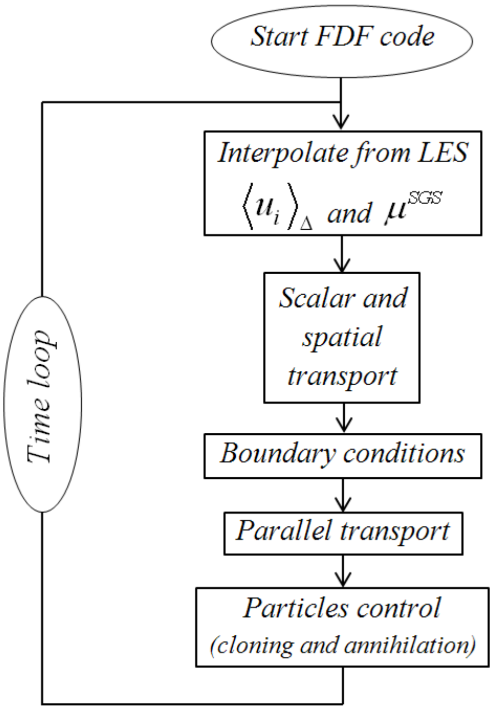

Start FDF code
Interpolate from LES
$\langle u_i \rangle_\Delta$ and μ^{SGS}
Scalar and spatial transport
Boundary conditions
Parallel transport
Particles control
(cloning and annihilation)
Time loop

Figure 4.1: Flowchart of the Lagrangian composition FDF code.

cell nodes to particle positions the filtered velocity $\langle u_i \rangle_\Delta$ and the SGS fluid viscosity μ^{SGS} — which will compose the SGS molecular diffusivity Γ^{SGS} — used in Eqs. (2.54) and (2.55).

Each particle evolves in Cartesian domain — spatial transport — and in composition space — scalar transport — by Monte Carlo solver. The code applies the outlet, inlet and/or wall-type boundary conditions, respectively, to delete particles from outflow regions, to feed particles on inflow regions or to reflect particles which intercepted a wall, with specular reflection. In distributed processing simulations, the algorithm also verifies in which process each particle dwells and transport it in groups among processes when necessary.

Population of particles constraint is the last step inside time loop. It involves statistical error management and controls memory / power processing usage by cloning and suppressing notional particles from all cells of flow domain.

4.3 Population of particles constraint

The FDF code must operate with a limited number of particles in whole flow domain N_{max} and with a bounded number of particle per cell ($NPC_{min} < NPC < NPC_{max}$). A common range, used in this master thesis, is $(0.8 \cdot NPC^0 < NPC < 1.5 \cdot NPC^0)$, where

NPC^0 is the initial number of particles per cell. The first constraint will depend on memory and power processing availability. Since each particle stands for a statistical sampling, the second restriction of the FDF code is concerned to statistical errors. The statistical error associated with the Monte Carlo technique is a function of NPC, and scales as $NPC^{-1/2}$ (HAWORTH, 2010).

Cloning and annihilation may alter particles composition and weight in order to conserve overall particles mass and to keep the ensemble mean and variance. While particles initial composition is just dependent on particle location, their initial weight is proportional to the host cell volume and inversely proportional to the number of particles inside it

$$w_p^* = \frac{\rho V}{NPC^0}, \tag{4.17}$$

where ρV is the fluid mass in cell and $p = 1, 2, ..., NPC^0$. Therefore, since fine and coarse cells have the same number of particles at initial stage, but the volume of the latter is eight times the volume of the former, particles from fine mesh regions have 1/8 statistical significance of those from coarse mesh regions.

4.3.1 Cloning

The cloning algorithm maintains a minimum number of notional particles inside each cell. If NPC becomes lesser than NPC_{min} in a given cell, the cloning procedure takes all particles of that cell and duplicates them, maintaining the clones with their original composition

$$\phi_{1'}^* = \phi_{2'}^* = \phi_1^*, \tag{4.18}$$

and with half of their original weight

$$w_{1'}^* = w_{2'}^* = \frac{1}{2} w_1^*, \tag{4.19}$$

where, composition $\phi_{n'}^*$ and weight $w_{n'}^*$ are concerned to new particles (n = 1, 2), and ϕ_1^* and w_1^* are concerned to the original ones.

This algorithm conserves the overall particles mass and keeps the FDF shape i.e., equal first and second order moments of composition.

Although cloned particles must continue inside the host cell, they may stand at the same position of the original one or shift their position randomly. The latter option helps to maintain particles uniform distribution, avoiding clustering. It has an important role on remeshing algorithm (section 4.5.2).

For each cell, the code accesses particle identifiers sequentially in hash table (chapter 3). Since particles cross cell borders randomly, given the nature of the Monte Carlo solver, we can consider particles selection in cloning procedure also a random process.

4.3.2 Annihilation

The annihilation algorithm keeps the number of stochastic particles in every cell below *a prior* defined limit NPC_{max}. Although larger set of statistical sampling produces better results, particles cloning without suppression may cause memory overload. If NPC becomes greater than NPC_{max} in a given cell, the annihilation procedure takes groups of 3 particles and suppresses them into groups of 2, according to

$$\phi^*_{1'} = \phi^*_{2'} = \frac{1}{3}\left(\phi^*_1 + \phi^*_2 + \phi^*_3\right), \tag{4.20}$$

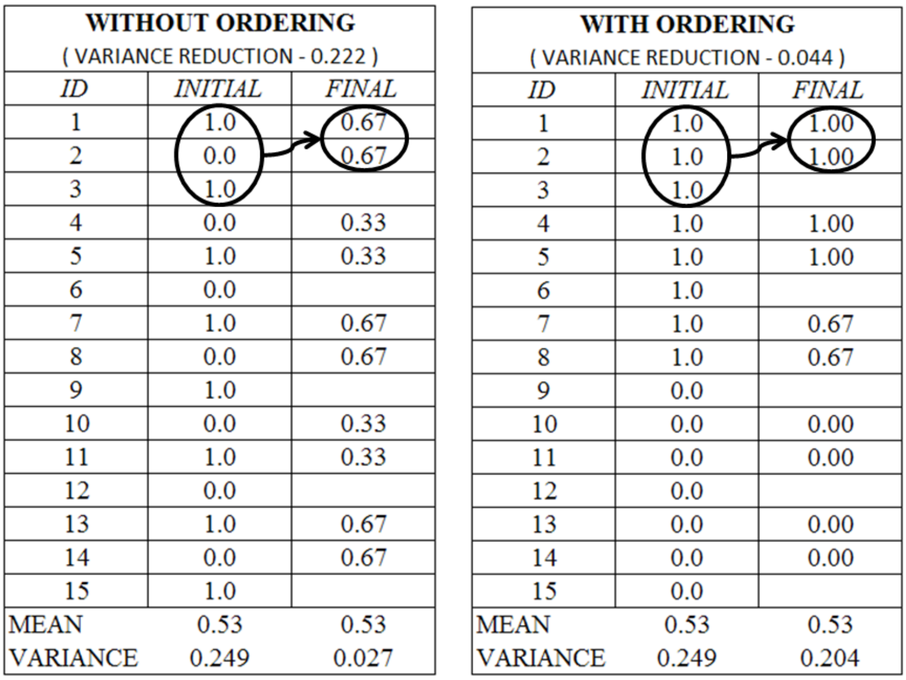
and,

$$w^*_{1'} = w^*_{2'} = \frac{1}{2}\left(w^*_1 + w^*_2 + w^*_3\right), \tag{4.21}$$

where, composition $\phi^*_{n'}$ and weight $w^*_{n'}$ are concerned to new particles ($n = 1,\,2$), and ϕ^*_m and w^*_m are concerned to the original ones ($m = 1,\,2,\,3$).

This algorithm conserves the overall particles mass and keeps the ensemble mean, but alters the composition second moment by artificial variance reduction. We can minimize this reduction ordering particles by their composition values Fig. (4.2). Haworth (2010) presents different annihilation forms.

Figure (4.2) shows two equal ensemble of 15 particles, before and after annihilation;



WITHOUT ORDERING (VARIANCE REDUCTION - 0.222)		
ID	*INITIAL*	*FINAL*
1	1.0	0.67
2	0.0	0.67
3	1.0	
4	0.0	0.33
5	1.0	0.33
6	0.0	
7	1.0	0.67
8	0.0	0.67
9	1.0	
10	0.0	0.33
11	1.0	0.33
12	0.0	
13	1.0	0.67
14	0.0	0.67
15	1.0	
MEAN	0.53	0.53
VARIANCE	0.249	0.027

(a)

WITH ORDERING (VARIANCE REDUCTION - 0.044)		
ID	*INITIAL*	*FINAL*
1	1.0	1.00
2	1.0	1.00
3	1.0	
4	1.0	1.00
5	1.0	1.00
6	1.0	
7	1.0	0.67
8	1.0	0.67
9	0.0	
10	0.0	0.00
11	0.0	0.00
12	0.0	
13	0.0	0.00
14	0.0	0.00
15	0.0	
MEAN	0.53	0.53
VARIANCE	0.249	0.204

(b)

Figure 4.2: Artificial variance reduction during annihilation.

just one is sorted by particles composition. Ordering does not affect the scalar mean of particles ensemble, instead it influences the scalar variance. Additionally, artificial mixing decreases as mixture advance.

4.4 Partially Stirred Reactor

The Partially Stirred Reactor (PaSR) simulation consists in calculating the effect of molecular diffusion over an initially segregated fluid using a micromixing model (ORBEGOSO, 2007). For industrial applications, we may model chemical reactors as PaSR. In CFD simulations that apply Lagrangian composition FDF method each cell is actually a PaSR.

In order to contextualize the CFD application of PaSR, we present the chemical reactor and some mathematical models. After, we show simulation results for a cubic PaSR using the Interaction by Exchange with the Mean (IEM) micromixing model.

4.4.1 Chemical reactors

Chemical reactors Fig. (4.3) are vessels designed to admit chemical reactions under defined process conditions e.g., pressure, temperature and concentrations. Some types of reactors have a stirrer attached in the main vessel in order to increase turbulence and, consequently, to improve mixing.

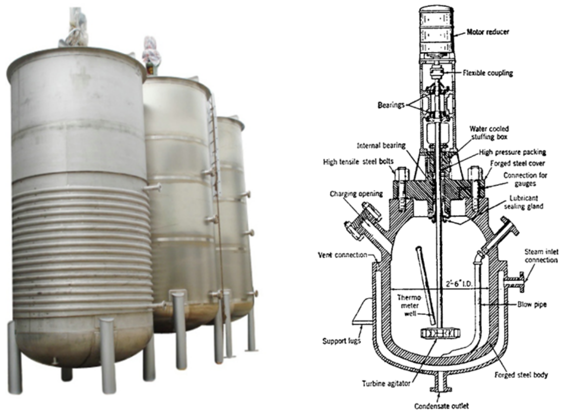


Figure 4.3: Industrial chemical reactor (left), and chemical reactor cross-section showing some of its components (right).

Mathematical models are necessary to calculate flow variables e.g., chemical species mass fractions. These models can be classified as Continuous Stirred-Tank Reactors (continuous operation) or Batch Reactors (intermittent operation).

4.4.2 Mathematical models

We may model the Continuous Stirred-Tank Reactor as a Perfectly Stirred Reactor (PSR), to which the hypothesis of fluid homogeneity apply. This hypothesis is based on the intense turbulent mixture inside the reactor. The PSR is not dependent on micromixing (i.e., effect of molecular diffusion on mixture), but just on fluid residence time. The assumption of PSR is not satisfied when the turbulent mixing rate is smaller than the chemical reaction rate, and the model becomes inaccurate to predict flow variables (ORBEGOSO, 2007).

On the other hand, the Plug Flow Reactor is the most recommended arrangement for slow mixture. In the middle of these two limits, we can apply the PaSR model (CORREA; BRAATEN, 1993).

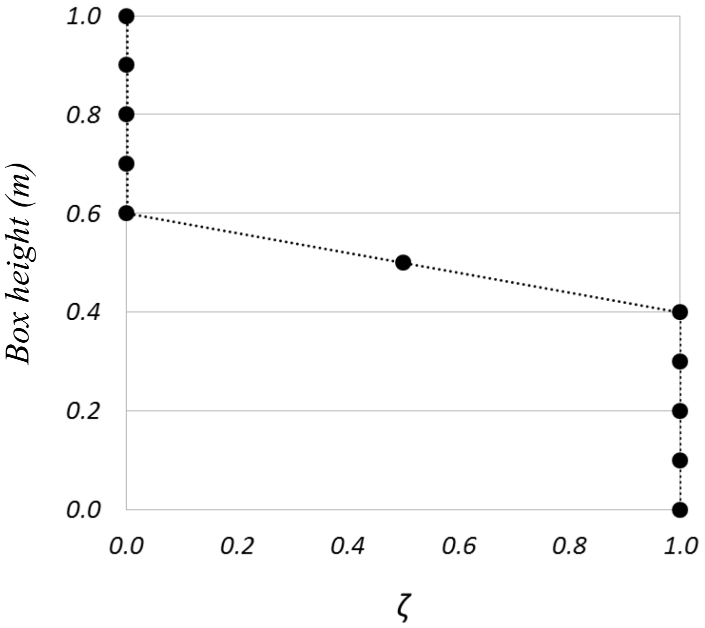


Figure 4.4: Initial scalar profile in PaSR. $\zeta_1 = 0$, $\zeta_2 = 1$, $\delta_m = 0.01$ and $h_m = 0.5$.

For reactive flows, the PaSR evaluates the micromixing influence over a chemical reaction. For instance, it is able to indicate the chemical kinetic response under the micromixing time, and when associated to thermal-kinetic mechanisms, leads to temperature and chemical species' mass fractions predictions (BASS; BARAT, 2003).

The IEM model is time dependent and provides PaSR micromixing. The composition of each notional particle reaches the global mean at each time step with a rate defined by the mixing frequency. This frequency corresponds to the turbulent dissipation and mixture energy ratio (BASS; BARAT, 2003).

4.4.3 PaSR simulation

The PaSR simulation is done in a cubic box, which is filled with notional particles that evolve in Cartesian and composition spaces Eqs. (2.54) and (2.55). The composition value of each particle is initialized by

$$\zeta\left(y\right) = \frac{\zeta_1 + \zeta_2}{2} + \frac{\zeta_1 - \zeta_2}{2} \tanh\left(\frac{2y}{\delta_m} - \frac{2h_m}{\delta_m}\right), \tag{4.22}$$

where, y is the box height coordinate, ζ_1 and ζ_2 are the initial composition values, δ_m is the initial mixture width and h_m is the coordinate where mixture initially occurs Fig. (4.4).

Figure (4.5) shows a particle tracking in x direction (box length). The Lagrangian element is initially at the center of the box, and keeps bounded in the range $y = [0.4 - 0.6]$ during all simulation according to Eq. (2.54).

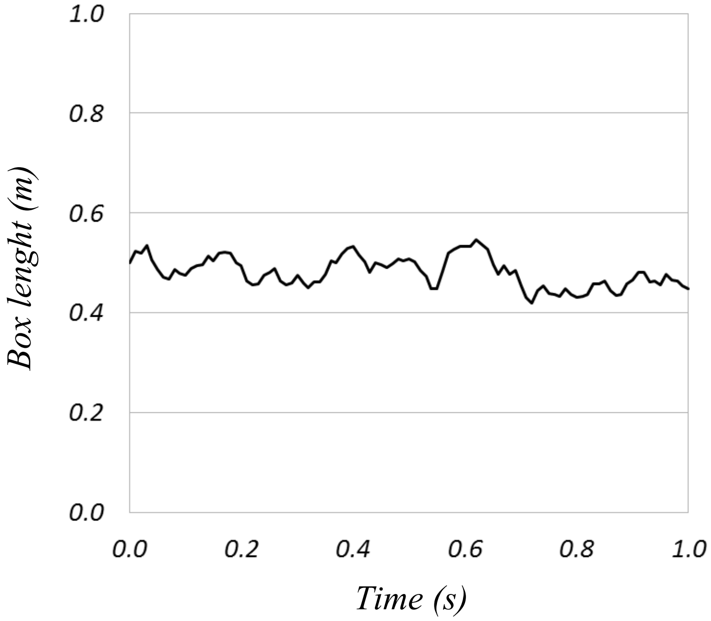


Figure 4.5: Particle tracking during the first second of PaSR simulation.

Figure (4.6) shows particles position and composition being changed by micromixing, according to Eq. (2.55). The majority of particles has $\phi^* = 0$ or $\phi^* = 1$ at first time; few particles appear in the mixture layer. Further, the composition of each particle tends to the global mean $\overline{\phi^*} \approx 0.45$.

The variable ζ describes an Eulerian composition field computed based on particles mean $\overline{\phi^*}$, and ϕ^* stands for a particle composition value.

The Probability Density Function (PDF) is defined as the distribution function derivative Eq. (2.21). It has wide usage in FDF methods that predict the value of SGS random variables in mixing and reactive flows (COLUCCI et al., 1998). The PDF is useful to analyze central moments (e.g., average) and distribution moments (e.g., variance) of a random variable, since these moments are integral functions of it.

In Lagrangian approach, each notional particle assumes a value between ϕ^*_{max} and ϕ^*_{min}. Therefore, we can discretize this range in order to count the number of particles that match each sub range, and further build the so called empirical PDF.

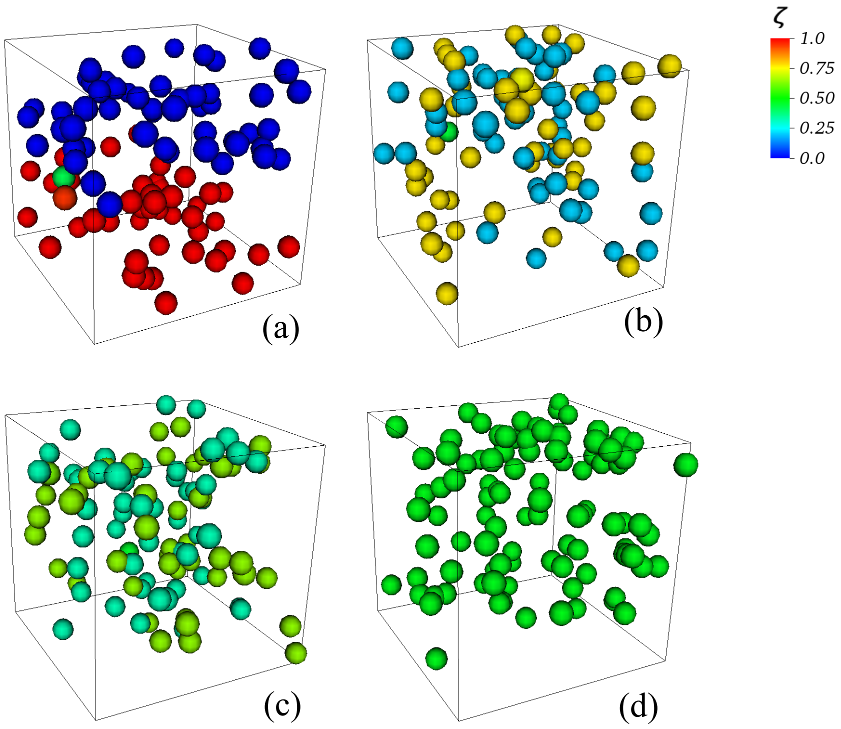


Figure 4.6: 3D view of PaSR with particles colored by composition. The blue and red colors represent the composition limits 0 and 1, respectively; the green color stands for the global mean.

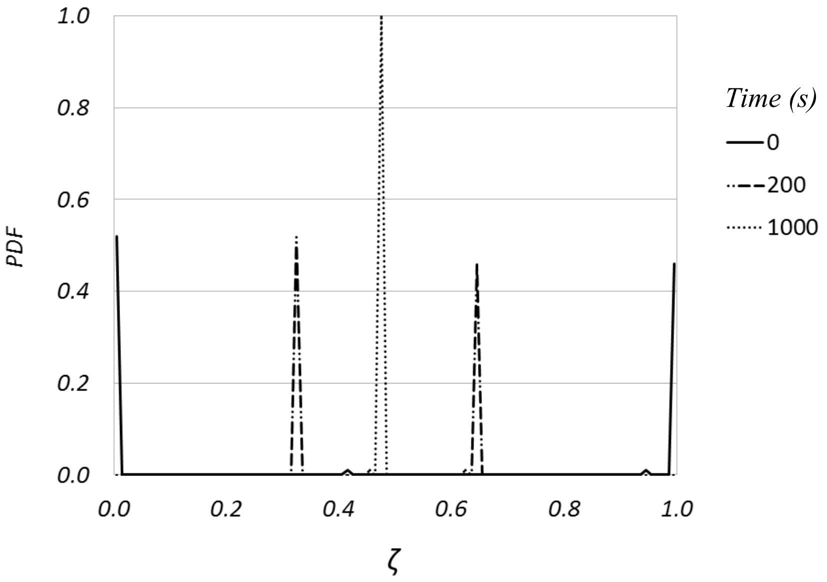


Figure 4.7: Empirical composition PDF during PaSR simulation.

Figure (4.7) presents the empirical PDF evolution in PaSR at three times: 0, 200 and 1000 s. The initial shape of double delta function (continuous line) tends to a single delta function at the global mean $\overline{\phi^*} \approx 0.45$ (dotted line).

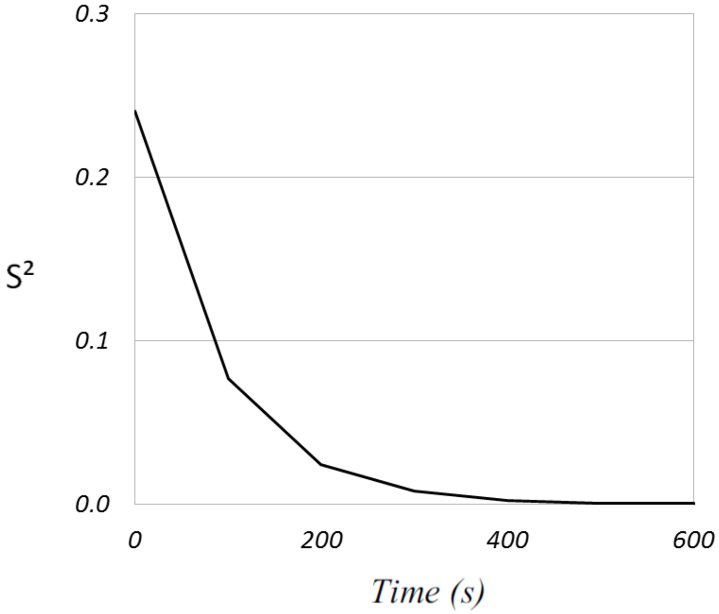


Figure 4.8: Composition variance during PaSR simulation.

As composition variance reduces, micromixing occurs (FOX, 2003). Figure (4.8) shows a line-graph that plots the composition variance against PaSR simulation time.

4.5 Statistical equivalence between Eulerian and Lagrangian approaches

The numerical solutions of Eqs. (2.3) or (2.14) by a Finite Differences method (Eulerian approach) and of Eqs. (2.54) and (2.55) by a Monte Carlo method (Lagrangian approach) must be statistically equivalent. It is a precondition of a hybrid LES / Lagrangian FDF method. As described in section 4.3, cloning and annihilation algorithms guarantee mass conservation of notional particles; it ensures the consistency of the numerical method (VEDOVOTO, 2011).

We present the PDE simulation cases in sections 4.5.1 and 4.5.2, for fixed and adaptive mesh simulations, respectively. They are based on FD solver for momentum and composition transport equations. The second order accuracy Central Difference and Crank-Nicolson numerical schemes integrate spatial and temporally the scalar transport equation, respectively.

For SDE simulation cases, we set the FD solver for momentum transport equation and the Lagrangian Monte Carlo solver for the composition transport by a system of stochastic differential equations. When applied to turbulent flows, this approach is called hybrid LES

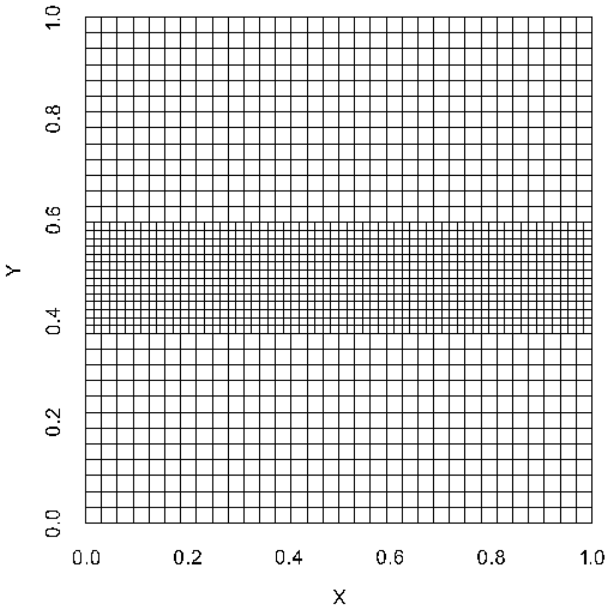


Figure 4.9: Multilevel, fixed mesh used on parallel, lid-driven cavity simulation. Longitudinal cross-section at $z = 0.5\,m$ — $\Delta^0 = 1/32$ and $\Delta^1 = 1/64$.

/ Lagrangian FDF method.

4.5.1 *Fixed mesh simulations*

We simulated a parallel lid-driven cavity with four processes aligned in x direction, in order to assess the statistical equivalence between PDE and SDE solutions. The main objectives were to analyze the influence of the number of notional particles NPC on SDE uniform mesh solution, and the effect of particles weight on composition field for multilevel, fixed mesh cases.

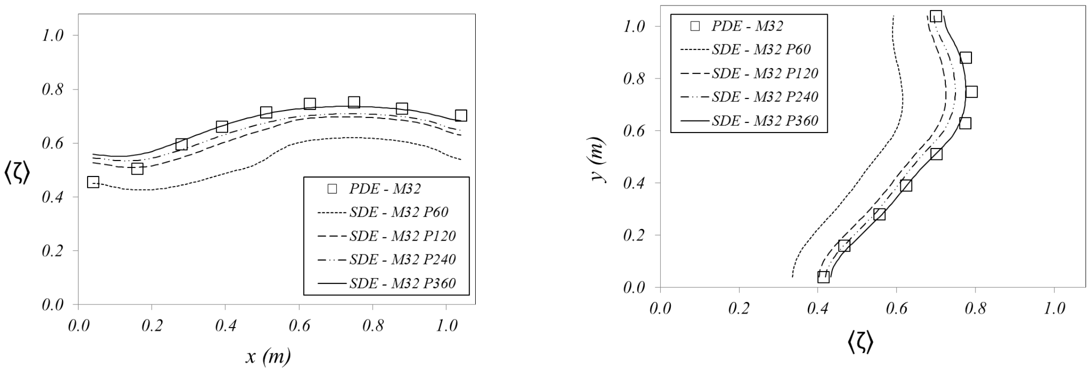
Table (4.1) presents the cases at $Re = 500$ run and their respective conditions. For all these simulations, the MC algorithm does not incorporate composition ordering in annihilation procedure (section 4.3.2).

In mixing flows into enclosed cavities, the overall composition mean $\bar{\zeta}$ (in this case equals $0.5\,(kg/kg)$) stands during all simulation, and the steady state of composition becomes a completely zero variance field.

The cubic cavity is unit volume and the flow is at $Re = 500$; the lid-driven is the north cavity wall (i.e., $y = 1.0\,m$). Additionally, we specified all simulations with Courant-Friedrichs-Lewy condition $CFL = 0.5$ and Schmidt number $Sc = 0.7$.

Table 4.1: Lid-driven cavity simulations on fixed, block-structured mesh.

Tag	Mesh type	Finest mesh width	NPC	Weighting
PDE - M32	Uniform	1/32	N/A	N/A
PDE - M64	Uniform	1/64	N/A	N/A
SDE - M32 P60	Uniform	1/32	60	No
SDE - M32 P120	Uniform	1/32	120	No
SDE - M32 P240	Uniform	1/32	240	No
SDE - M32 P360	Uniform	1/32	360	No
SDE - MG64 P60	Fixed MG	1/64	60	No
SDE - MG64 P240	Fixed MG	1/64	240	No
SDE - MG64 P240 W	Fixed MG	1/64	240	Yes



(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

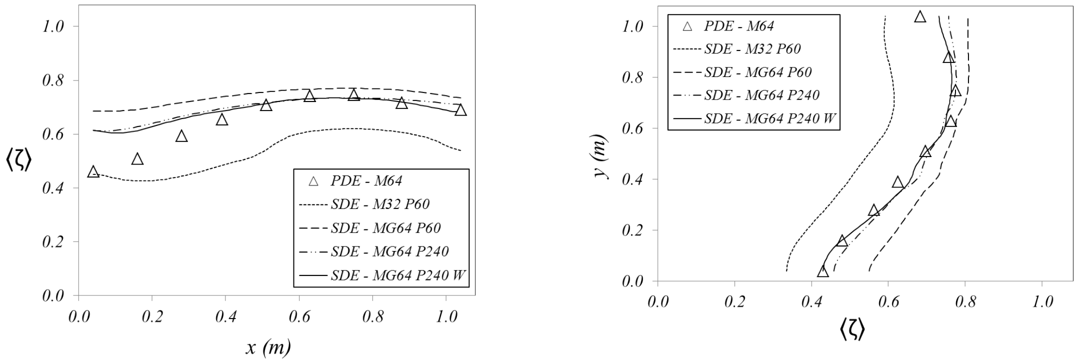
Figure 4.10: Profiles of temporal composition average — PDE and SDE simulations at $t = 30.0\,s$ and $Re = 500$ on uniform mesh with 32^3 cells . NPC ranges from 60 to 360.

For multilevel, fixed mesh cases, we set the finest mesh as a $x - line$ strip at $y = 0.5\,m$ and throughout XZ plan Fig. (4.9). Thus, every $x - line$ profiles presented in following results dwell entirely in this region.

A particle composition value follows Eq. (4.22) in order to produce a segregated initial composition field. We took the temporal composition average $\langle\zeta\rangle$ from $t = 0.1\,s$ to $t = 30.0\,s$ for these simulations.

Figure (4.10) shows two temporal composition average profiles to compare the PDE and SDE cases on uniform mesh. In these simulations, there is no difference from to calculate composition statistics pondered by notional particles weight or not. It happens because all cells have the same size in flow domain and the fluid is unit density.

As NPC increases, the SDE profiles tend to match the PDE result. We obtained the

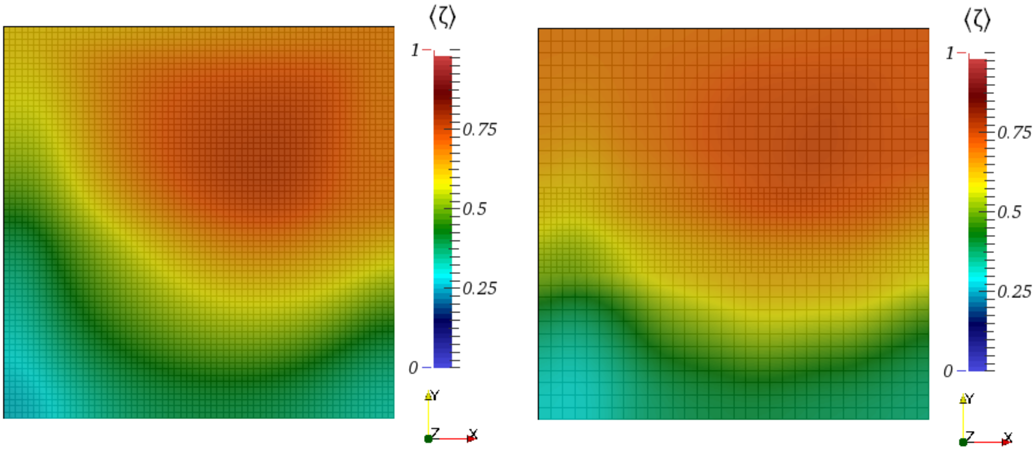


(a) $x - line$ at $y = z = 0.5\,m$ (b) $y - line$ at $x = z = 0.5\,m$

Figure 4.11: Profiles of temporal composition average — PDE and SDE simulations at $t = 30.0\,s$ and $Re = 500$ on uniform and multilevel, fixed meshes.

greatest improvement on SDE solution doubling NPC from 60 to 120; after these numbers, minor differences occurred. For an uniform mesh with 32^3 cells, SDE results initially overpredicted mixture, presenting lower composition values.

Simulations using multilevel, fixed mesh require to calculate composition statistics pondered by notional particles weight. Because the weight is proportional to the cell volume Eq. (4.17), particles which dwell in fine mesh regions have lesser statistical significance than those in coarse mesh regions.



(a) PDE (b) SDE

Figure 4.12: Comparison between PDE and SDE temporal composition average fields at $t = 30.0\,s$, $Re = 500$ and $z = 0.5\,m$.

Figure (4.11) shows two temporal composition average profiles to compare the PDE and SDE cases on multilevel, fixed mesh. The first improvement occurred when we placed a fine mesh $\Delta^1 = 1/64$ in the range $y = [0.4 - 0.6]$ throughout XZ plan. This new multilevel, fixed mesh initially under-predicted mixing, presenting higher composition values.

Both $x-line$ and $y-line$ profiles of SDE cases improve when NPC increases, and also by calculating composition statistics pondered by particles weight. The $SDE - M32\,P60$ and $SDE - MG64\,P240\,W$ stand for a 60 particles per cell simulation on uniform mesh with 32^3 cells and a 240 particles per cell, weight pondered simulation on multilevel, fixed mesh Fig. (4.9), respectively.

Figure (4.12) shows the PDE (a) and the SDE (b) temporal composition average fields computed from $t = 0.1\,s$ to $t = 30.0\,s$. The PDE simulation was based on uniform mesh with 64^3 cells, and the SDE simulation is $MG64\,P240\,W$. It shows the SDE solution under-predicted composition field near the west cavity wall (i.e., $x = 0.0\,m$).

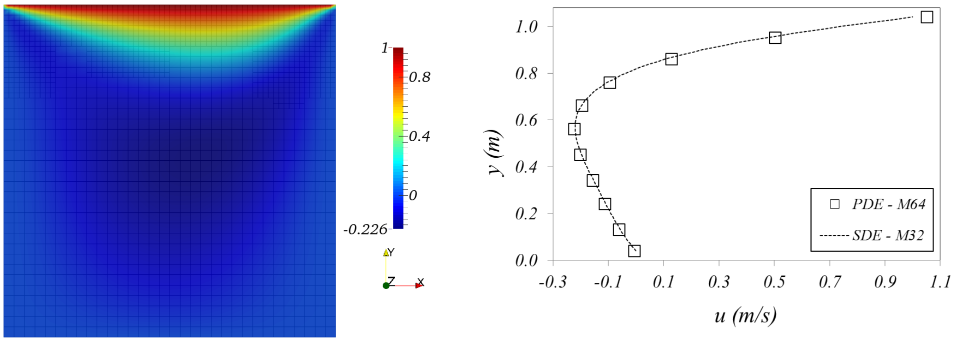
When the algorithm does not suppress notional particles ordered by their composition values, it demands more NPC to achieve good results. The next section shows simulations that incorporated this sorting procedure.

4.5.2 Adaptive mesh simulations

Adaptive mesh differs from fixed one due to the need to exchange flow variables from the old to the new grid after remeshing. In the Lagrangian framework, all notional particles must find their address in the new Lagrangian map every time remeshing occurs. The AMR provides high numerical precision on flow regions of interest, for instance on extreme gradient zones.

When a coarse cell is refined, each one of the eight offspring cells inherits a fraction of particles from it. Further, the cloning procedure operates upon these cells, which certainly have $NPC < NPC_{min}$. On the other hand, when eight fine cells cluster forming one coarse cell, this cell inherits all particles from the eight ancestors. Further, the annihilation procedure operates upon this cell, which certainly has $NPC > NPC_{max}$.

This section covers two Reynolds number, laminar $Re = 50$ and turbulent $Re = 5000$, in a parallel, lid-driven cavity simulation using adaptive, block-structured mesh with four processes aligned in x direction. The lid-driven is the north cavity wall (i.e., $y = 1.0\,m$). For



(a) $x - velocity$ field

(b) $x - velocity$ profiles

Figure 4.13: (a) Longitudinal cross-section of $x - velocity$ field at $t = 6.0\,s$, $Re = 50$ and $z = 0.5\,m$, and (b) profiles of PDE and SDE solutions at $x = 0.5\,m$.

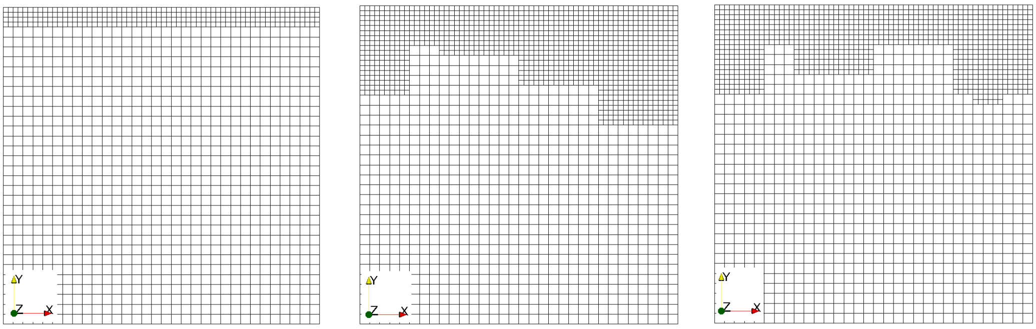
laminar and turbulent flows simulations, we assessed the composition ordering influence on SDE results and analyzed the effect of SGS convective flux Eq. (2.46) on mixing, respectively. The algorithm comprises Eq. (2.55) and the velocity and the Wiener noise terms of Eq. (2.54) in laminar flow simulations. Table (4.2) presents the cases run and their respective conditions.

Table 4.2: Lid-driven cavity simulations on adaptive, block-structured mesh.

Tag	Re	Finest mesh width	NPC	Particles sorting	SGS convective flux
PDE - M32	50	1/32	N/A	N/A	N/A
PDE - M64	50	1/64	N/A	N/A	N/A
SDE - D30	50	1/64	30	No	N/A
SDE - O30	50	1/64	30	Yes	N/A
SDE - O60	50	1/64	60	Yes	N/A
SDE - LS	5000	1/64	60	Yes	No
SDE - SGS	5000	1/64	60	Yes	Yes

The particles composition value follows Eq. (4.22) in order to produce an initially segregated composition field. We took the temporal composition average $\langle \zeta \rangle$ from $t = 1.0\,s$ to $t = 6.0\,s$ for simulations at $Re = 50$ and from $t = 1.0\,s$ to $t = 24.0\,s$ for simulations at $Re = 5000$.

Since PDE cases are benchmarks, they were performed on a uniform mesh with 64^3 cells. It was the highest mesh resolution we applied in simulations using an Intel® Core™ i5-3450, 3.10 GHz CPU, 1,600.00 MHz speed computer. For SDE cases, we set AMR with



(a) initial time (b) intermediate time (c) final time, $t = 6.0\,s$

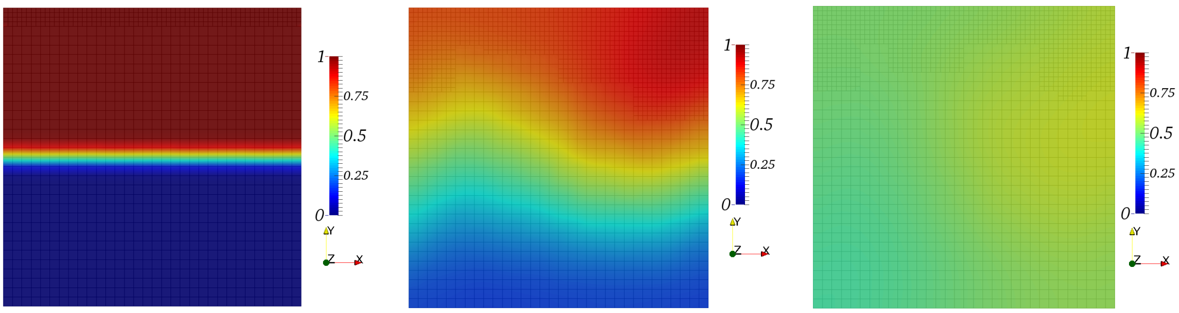
Figure 4.14: Longitudinal cross-section of AMR — SDE cases at $Re = 50$ and $z = 0.5\,m$.

two mesh levels: $\Delta^0 = 1/32$ and $\Delta^1 = 1/64$. Additionally, we specified all simulations with Courant–Friedrichs–Lewy condition $CFL = 0.5$ and Schmidt number $Sc = 0.7$.

Laminar flows

Figure (4.13) shows the $x - velocity$ field with PDE and SDE profiles as a function of y at $x = 0.5\,m$ for laminar flow. Since these profiles agreed with each other, we can isolate the influence of advection from diffusion modeling when comparing composition fields. The $y - velocity$ and $z - velocity$ presented the same pattern of $x - velocity$. In fact, we expected it because the numerical model used to solve the momentum equations was the same for both PDE and SDE cases.

Figure (4.14) shows three time steps of mesh longitudinal cross-section at $z = 0.5\,m$ of SDE cases. Although the initial gradient of composition field is in the middle of the cavity, the refinement follows vorticity, and therefore began near the lid-driven (i.e., $y = 1.0\,m$).

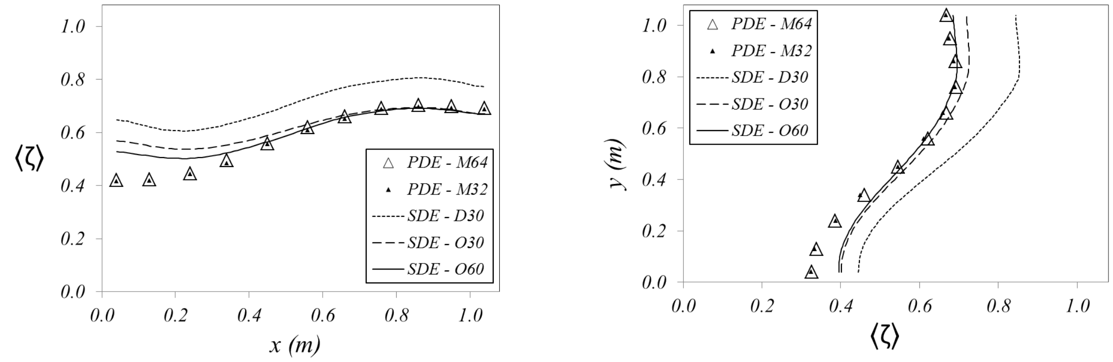


(a) initial time (b) intermediate time (c) final time, $t = 6.0\,s$

Figure 4.15: Longitudinal cross-section of temporal composition average $\langle\zeta\rangle$ — $SDE - O60$ simulation at $Re = 50$ and $z = 0.5\,m$.

Figure (4.15) shows three time steps of temporal composition average $\langle\zeta\rangle$ longitudinal cross-section at $z = 0.5\,m$ of $SDE - O60$ simulation.

The initially segregated fluid inside the lid-driven cavity began mixing in the boundary of the composition gradient, which deforms due to convection. The temporal composition average tends to a homogeneous $\langle\zeta\rangle = 0.5\,kg/kg$, zero variance field, represented by the green color.



(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

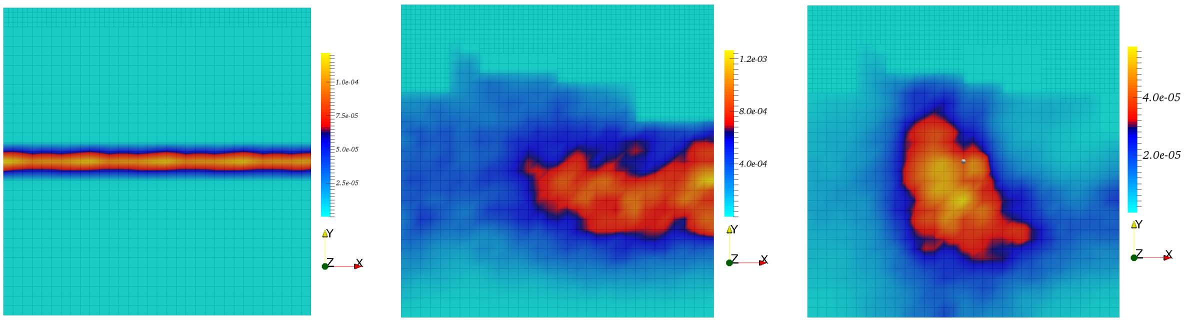
Figure 4.16: Profiles of temporal composition average at $t = 6.0\,s$ and $Re = 50$.

Figure (4.16) shows the PDE and SDE profiles to compare the temporal composition average fields. The $SDE - O30$ and $SDE - D30$ cases stand for simulations with and without composition ordering, respectively. We obtained better results when the annihilation procedure had sorted particles in each cell according with their composition value. An improvement occurred by doubling NPC from 30 to 60; the PDE and SDE profiles tend to entirely agree increasing NPC. Moreover, the used mesh initially under-predicted mixing, presenting higher composition values.

In the $SDE - D30$ case, the algorithm iterates hash table directly and, due to the MC solver nature, we can consider the selection of particles a random process.

Figure (4.17) shows three time steps of composition variance S^2 longitudinal cross-section at $z = 0.5\,m$ of $SDE - O60$ simulation. At first time step, the variance field has a maximum value of about $1 \cdot 10^{-4}\,(kg/kg)^2$. This value increases up to about $1 \cdot 10^{-3}\,(kg/kg)^2$ during simulation, and then decreases to about $1 \cdot 10^{-5}\,(kg/kg)^2$ at $6.0\,s$.

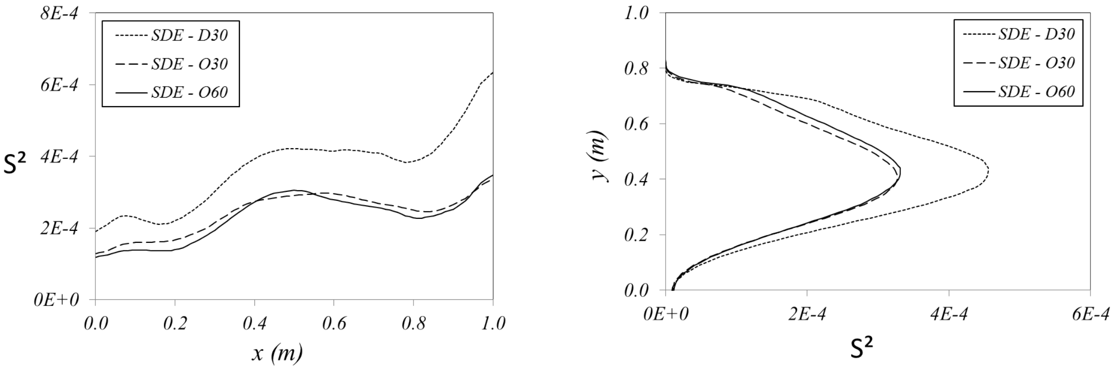
Figure (4.18) shows the composition variance S^2 profiles. Although the $SDE - O30$



(a) initial time (b) intermediate time (c) final time, $t = 6.0\,s$

Figure 4.17: Longitudinal cross-section of variance S^2 calculated based on temporal composition average — $SDE - O60$ simulation at $Re = 50$ and $z = 0.5\,m$.

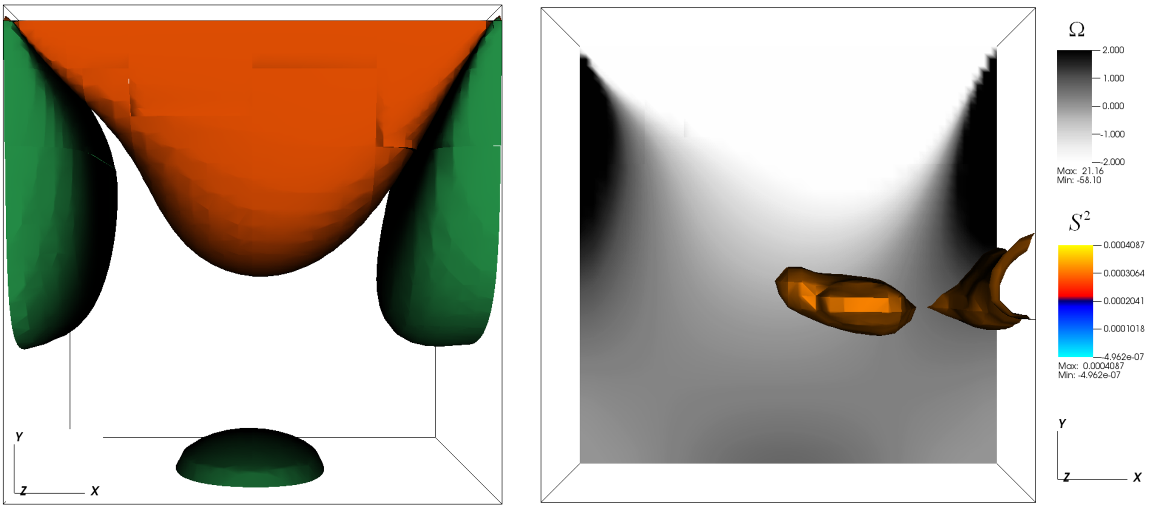
and $SDE - O60$ simulations presented differences in composition field, their variance fields are similar. However, their results differ about 30% at $y = 0.4\,m$ from the $SDE - D30$ result, revealing that annihilation ordering is more significative than number of particles per cell.



(a) $x - line$ at $y = z = 0.5\,m$ (b) $y - line$ at $x = z = 0.5\,m$

Figure 4.18: Profiles of composition variance calculated based on temporal composition average at $t = 6.0\,s$ and $Re = 50$.

The composition variance field reveals where mixture has taken place; in other words, where fluids with different composition form interfaces enhancing molecular diffusion. It initially occurs in the middle of the flow domain due to the initial condition of composition; further, it shifts to the east wall (i.e., $x = 1.0\,m$) and stands there forced by two opposite vortex in z direction; finally, the mixture zone detaches from the wall and transforms into a spherical like region in the midst of the cavity Fig. (4.17). Figure (4.19) shows the relationship between composition variance and vorticity field.



(a) (b)

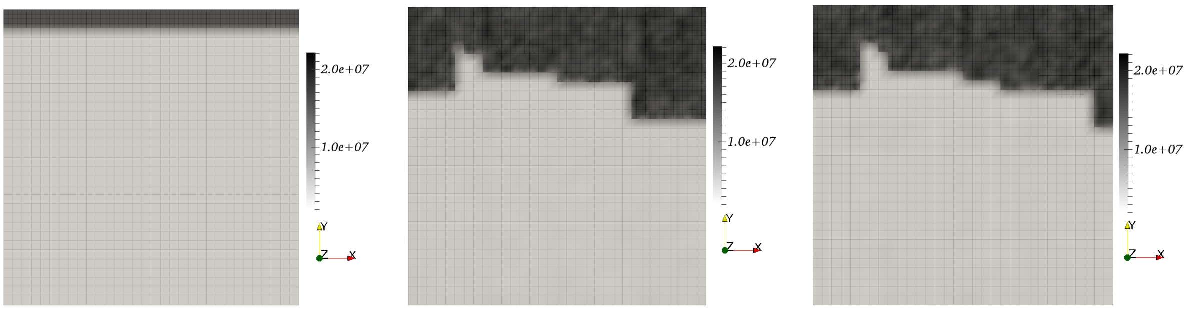
Figure 4.19: Relationship between composition variance and vorticity at $t = 6.0\,s$ and $Re = 50$. (a) Isosurface $|z - vorticity| = 0.5\,s^{-1}$ — clockwise/negative (orange) and counterclockwise/positive (green). (b) Longitudinal cross-section of $|z - vorticity|$ and 3D view of composition variance $S^2 = 3 \cdot 10^{-4}\,(kg/kg)^2$.

Due to cloning and annihilation procedures the Monte Carlo algorithm remains under a controlled population of particles inside each cell i.e., from the first time step to the last one, NPC keeps inside the range of $[50 - 90]$.

Figure (4.20) shows three time steps of particle density field longitudinal cross-section at $z = 0.5\,m$ of $SDE - O60$ simulation. Since NPC^0 is independent of cell size, the particle density (NPC/m^3) is higher in fine grid cells than in coarse ones. The statistical error stands homogeneous throughout flow domain; it is dependent on the number of particle per cell, instead of the particle density. Nonetheless, the particle density serves to indicate computational effort and requested memory in each computer process.

Turbulent flows

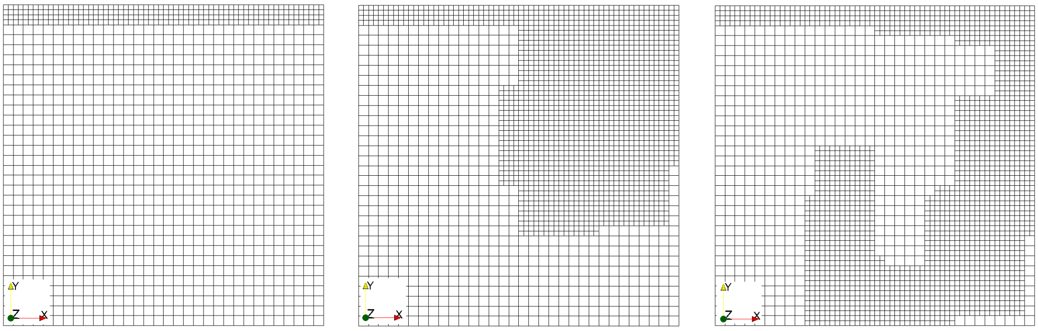
We simulated a two-species, lid-driven cavity, turbulent flow in order to measure the influence of SGS convective flux on mixture — the macromixing effect (HAWORTH, 2010). This is the unique simulation in this master thesis which completely demanded the implemented hybrid LES / Lagrangian composition FDF method. In other words, this case covers



(a) initial time (b) intermediate time (c) final time, $t = 6.0\,s$

Figure 4.20: Longitudinal cross-section of notional particle density (NPC/m^3) — $SDE{-}O60$ simulation at $Re = 50$ and $z = 0.5\,m$.

the whole flowchart of Fig. (4.1) and used all terms of Eqs. (2.54) and (2.55). The SGS (subgrid scale) and LS (large scale) cases stand for mathematical modeling with and without the SGS convective flux term, respectively.



(a) initial time (b) intermediate time (c) final time, $t = 24.0\,s$

Figure 4.21: Longitudinal cross-section of AMR — SGS cases at $Re = 5000$ and $z = 0.5\,m$.

Figure (4.21) shows three time steps of mesh longitudinal cross-section at $z = 0.5\,m$ of turbulent flow cases. As happened with laminar flow, the refinement began near the lid-driven (i.e., $y = 1.0\,m$) and follows vorticity. However, the refinement covers a wider region in turbulent than in laminar flow, reaching the bottom of the cavity in the former case.

The turbulence in the flow creates smaller kinematic and composition structures than the mesh resolution of Fig. (4.21). Hence, the numerical model solves part of the momentum field and models other part; the same occurs to the composition field, see chapter 2.2.

Figure (4.22) shows turbulence length scales ranging from Integral to Kolmogorov scales, and presents the relative position of Δ^0 and Δ^1 grid resolutions, above which the numerical model solves the flow structures and below which it models them.

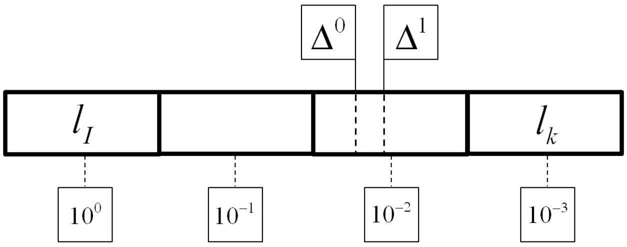


Figure 4.22: Integral to Kolmogorov turbulence length scales, $Re = 5000$.

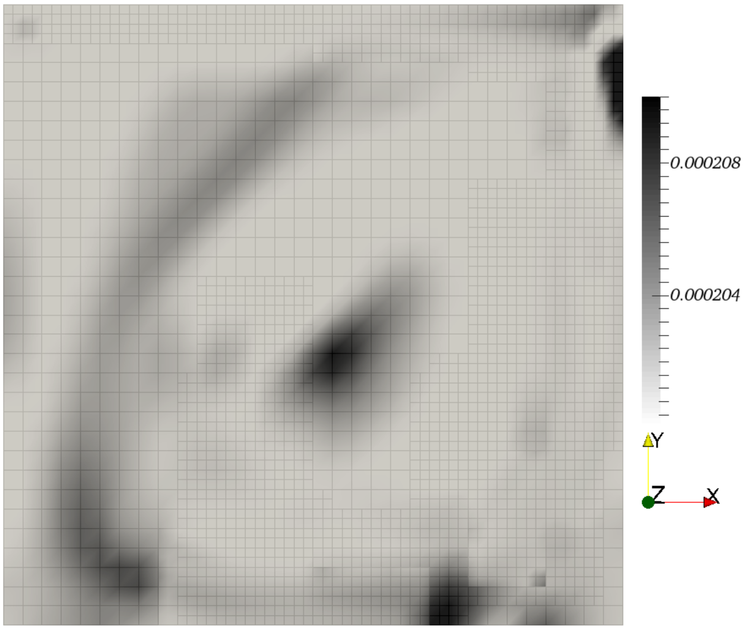
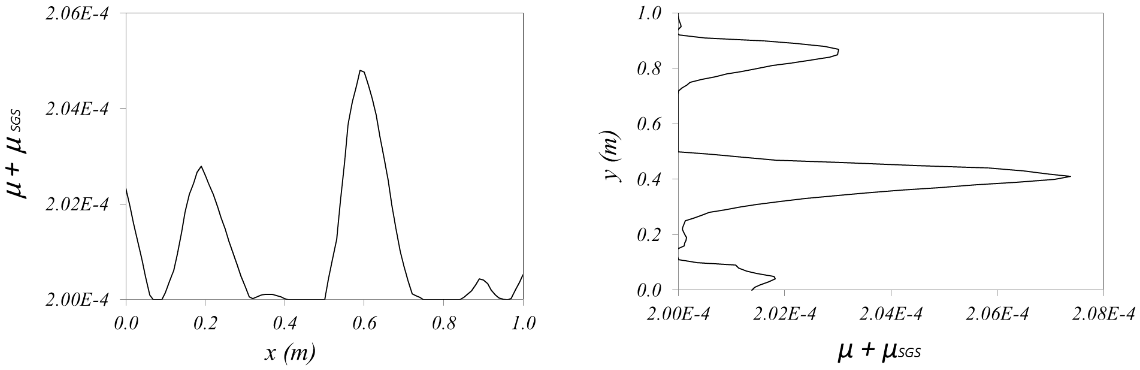


Figure 4.23: Longitudinal cross-section of effective dynamic viscosity $\mu + \mu^{SGS}$ — SGS simulation at $t = 24.0\,s$, $Re = 5000$ and $z = 0.5\,m$.



(a) $x - line$ at $y = z = 0.5\,m$

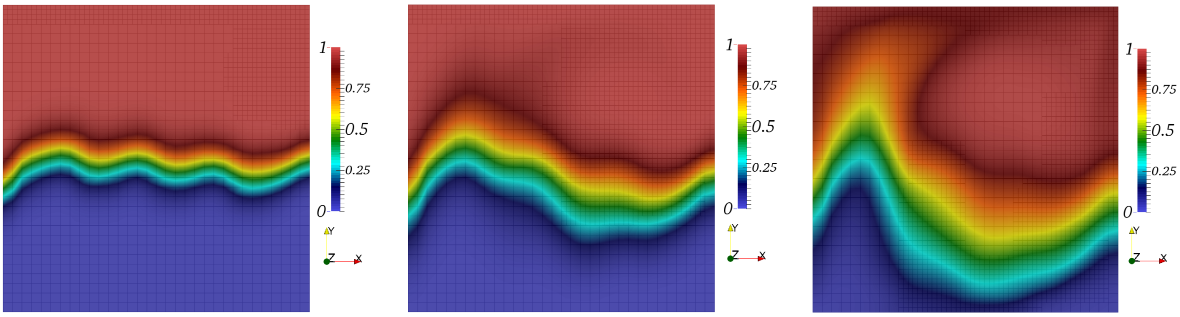
(b) $y - line$ at $x = z = 0.5\,m$

Figure 4.24: Profiles of effective dynamic viscosity at $24.0\,s$ and $Re = 5000$.

The Kolmogorov length scale l_k represents the length of the smallest kinematic structure in turbulent flow. It relates to the Integral length scale l_I and Reynolds number by

$$l_k = \frac{l_I}{\mathrm{Re}^{3/4}}. \tag{4.23}$$

The smallest composition length scale relates to the Kolmogorov scale by the Schmidt number, which was set to $Sc = 0.7$ in all cases we simulated.



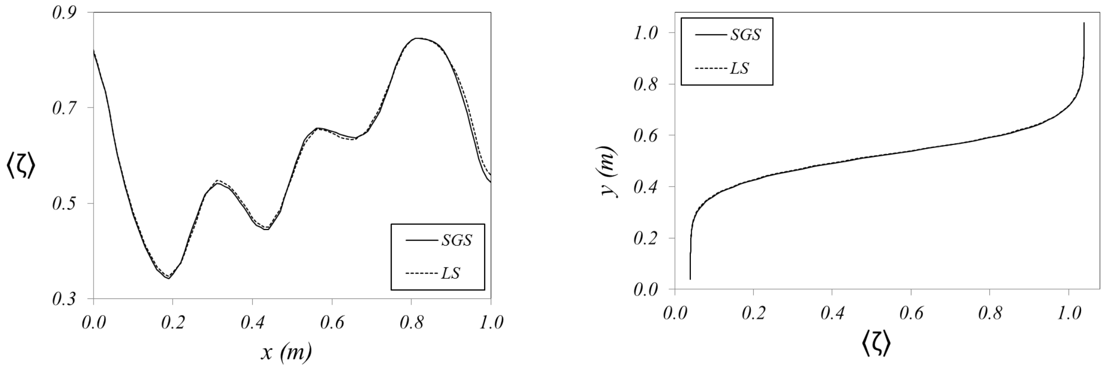
(a) initial time

(b) intermediate time

(c) final time, $t = 24.0\,s$

Figure 4.25: Longitudinal cross-section of temporal composition average $\langle\zeta\rangle$ — SGS simulation at $Re = 5000$ and $z = 0.5\,m$.

Figure (4.23) shows mesh resolution of Fig. (4.21) applied to a flow at $Re = 5000$ creates a SGS dynamic viscosity field with maximum value lesser than 5% of the molecular dynamic viscosity $2 \cdot 10^{-4}\,Pa \cdot s$ at $z = 0.5\,m$ using a turbulence dynamic model (MELO et al., 2014). Figure (4.24) shows two effective dynamic viscosity profiles, which reveals the highest μ^{SGS} values are on coarse mesh Δ^0, as expected.

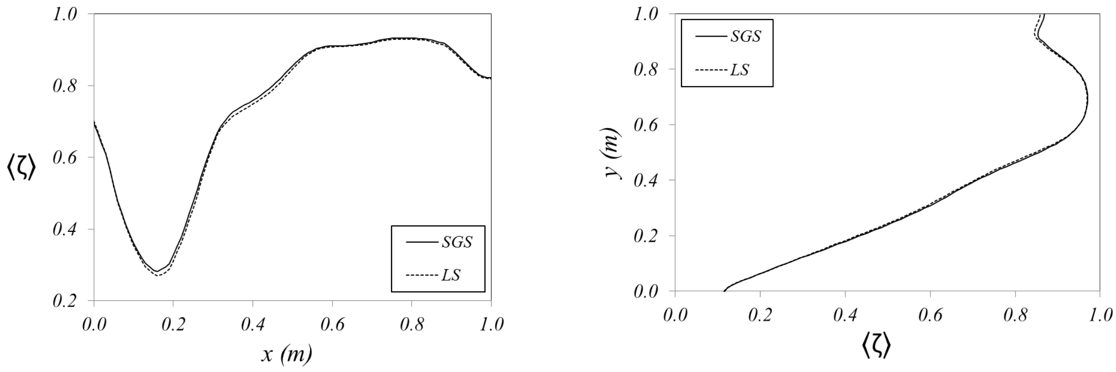


(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

Figure 4.26: Profiles of temporal composition average at $t = 6.0\,s$ and $Re = 5000$.

As occurred for laminar flow, the initially segregated fluid inside the lid-driven cavity begins to mix in the boundary of composition gradient, which deforms due to convection; the green color represents the mixing zone. Figure (4.25) shows three time steps of temporal composition average $\langle\zeta\rangle$ longitudinal cross-section at $z = 0.5\,m$ of SGS simulation.

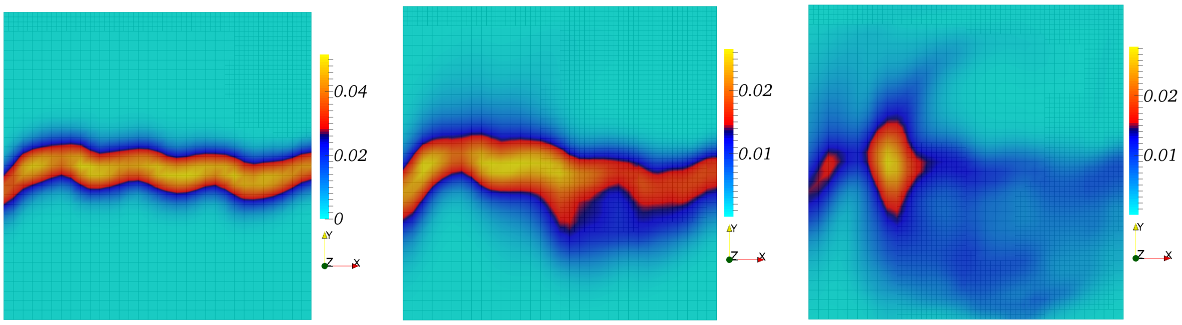


(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

Figure 4.27: Profiles of temporal composition average at $t = 24.0\,s$ and $Re = 5000$.

Figures (4.26) and (4.27) show the SGS and LS profiles to compare the temporal composition average fields at $6.0\,s$ and $24.0\,s$, respectively. For both cases, the turbulence level is not enough to produce different profiles. In conclusion, macromixing modeling has small effect in the lid-driven cavity flow simulation at $Re = 5000$ with the used mesh.



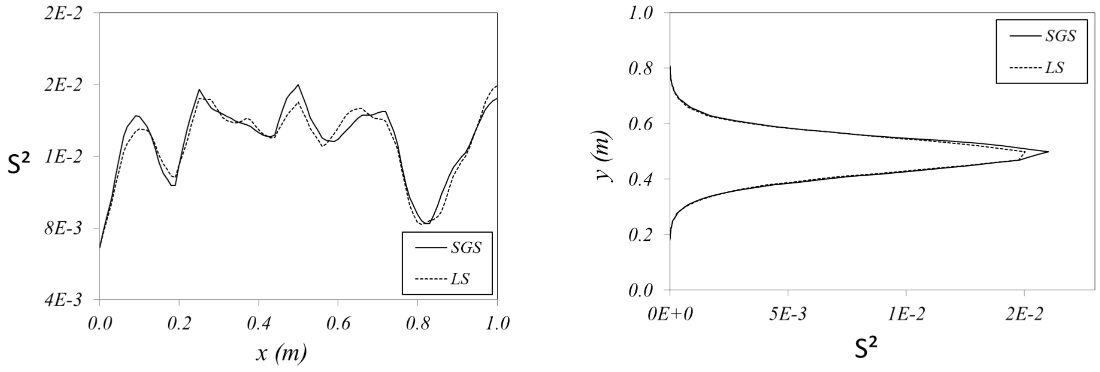
(a) initial time

(b) intermediate time

(c) final time, $t = 24.0\,s$

Figure 4.28: Longitudinal cross-section of variance S^2 calculated based on temporal composition average — SGS simulation at $Re = 5000$ and $z = 0.5\,m$.

At first time step, the variance field has a maximum value of about $4 \cdot 10^{-2}\,(kg/kg)^2$. This value decreases to about $2 \cdot 10^{-2}\,(kg/kg)^2$ during simulation, and keeps it until the final simulation time. Figure (4.28) shows three time steps of composition variance S^2 longitudinal cross-section at $z = 0.5\,m$ of SGS simulation.

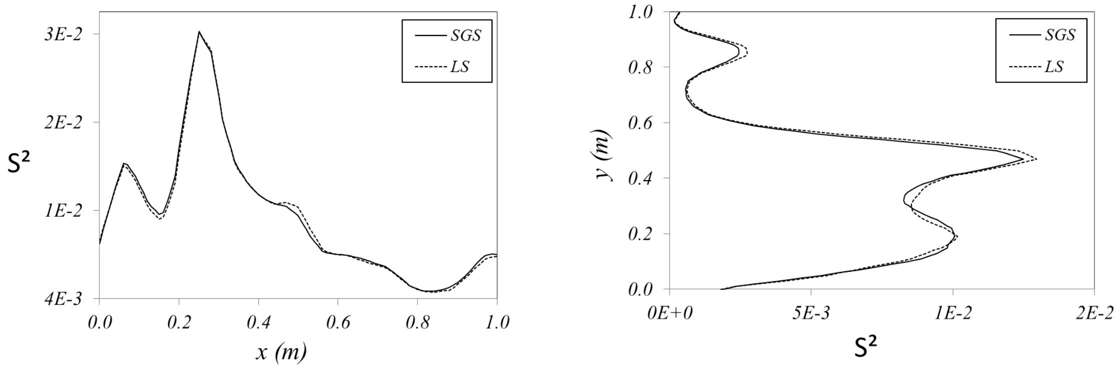


(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

Figure 4.29: Profiles of composition variance calculated based on temporal composition average at $t = 6.0\,s$ and $Re = 5000$.

Figures (4.29) and (4.30) show SGS and LS profiles to compare the composition variance fields at 6.0 s and 24.0 s, respectively. As temporal composition average profiles revealed, S^2 profiles also show SGS simulation presents no difference from LS simulation.

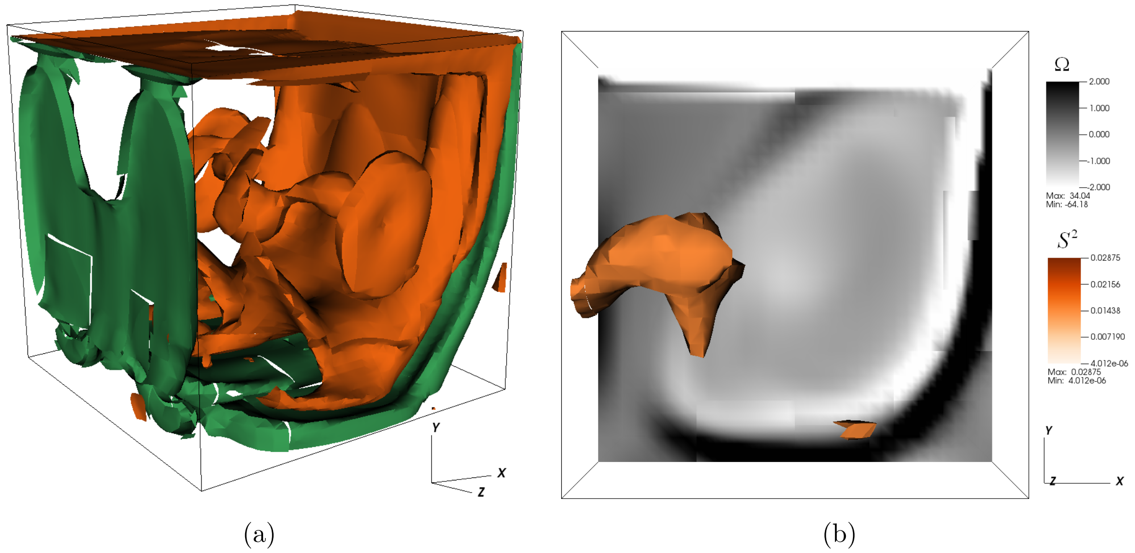


(a) $x - line$ at $y = z = 0.5\,m$

(b) $y - line$ at $x = z = 0.5\,m$

Figure 4.30: Profiles of composition variance calculated based on temporal composition average at $t = 24.0\,s$ and $Re = 5000$.

For turbulent mixing flow, composition mixing initially occurred in the middle of flow domain due to the initial condition of composition, but further shifts to the west wall (i.e., $x = 0.0\,m$) moved by a complex vortex structures in z direction. Figure (4.31) shows a higher level of mixing occurs in turbulent flow at $Re = 5000$ $S^2 \approx 10^{-2}\,(kg/kg)^2$ compared to laminar flow at $Re = 50$ $S^2 \approx 10^{-4}\,(kg/kg)^2$, demonstrating a proportionality between Reynolds number and mixing extent.



(a) (b)

Figure 4.31: Relationship between composition variance and vorticity at $t = 24.0\,s$ and $Re = 5000$. (a) Isosurface $|z - vorticity| = 0.5\,s^{-1}$ — clockwise/negative (orange) and counterclockwise/positive (green). (b) Longitudinal cross-section $|z - vorticity|$ and 3D view of composition variance $S^2 = 2 \cdot 10^{-2}\,(kg/kg)^2$.

CHAPTER V

CONCLUDING REMARKS

We fulfilled the proposed goals for this master thesis by developing a computational model on a distributed processing environment, implementing a hybrid LES / Lagrangian FDF method, applying the Monte Carlo technique on adaptive, block-structured mesh and simulating laminar and turbulent mixing flows in order to check the consistency of the stochastic method.

A parallel Lagrangian framework was developed, which demanded:

- Creating the Lagrangian map composed by a multi-level hash table;
- Expanding the serial Lagrangian map to a parallel version;
- Merging it with the in-house built AMR3D parallel code, and establishing the communication between Fortran and C based functions. We used these programming languages to develop Eulerian and Lagrangian based codes, respectively.

These developments brought the following benefits to the AMR3D code:

- It is a useful framework to implement a hybrid LES / Lagrangian composition FDF method;
- The algorithm guarantees a feasible and quick way to handle the parallel transport of any kind of particle on the complex adaptive, block-structured mesh;

- The hash table provides direct access to the ensemble of particles inside each Eulerian cell;
- The developed particle searching algorithm is mesh size independent;
- The computational model optimizes the tasks of the Monte Carlo solver and, consequently, the simulation time;
- The Lagrangian framework allows tracking different types of Lagrangian elements and relating them e.g., notional particles which are going to cross a plane defined by Front Tracking nodes.

The transport of particles on distributed processing environment was developed using the Message Passing Interface (MPI), which allows:

- Parallel simulation of flows using Lagrangian stochastic methods;
- Parallel simulation of flows involving particle modeling, such as gas-solid flows;
- Front Tracking nodes management on distributed processing applications.

A hybrid Large Eddy Simulation (LES) / Lagrangian composition Filtered Density Function (FDF) method on adaptive, block-structured mesh (AMR) was implemented. This method has a potential to:

- Simulate the physics of multiple-species, turbulent mixing flows with chemical reaction and combustion, using the LES approach and modeling the subgrid composition fluctuations in a probabilistic manner;
- Directly model the chemical reaction rate.

A Monte Carlo technique on adaptive, block-structured mesh was developed. This involves notional particles normalization and comprehends the processes of:

- Notional particles cloning, guarantying a bounded Monte Carlo accuracy and maintaining the numerical error homogeneously distributed throughout flow domain;

- Notional particles annihilation, preventing simulation from computer memory overload and handling the numerical error.

The MC algorithm on AMR deals with a great cell volume change during the refinement process i.e., 8 times; yet it keeps the number of notional particles per cell into a fixed range. This algorithm also confers high numerical resolution on critical regions of the flow, such as regions with sharp variations in composition field; in other words, although the accuracy is equal for every cell, there are more cells per flow volume.

The laminar and turbulent, non-reactive, two-species mixing flows with unit density and constant fluid properties in a lid-driven cavity were simulated using both the developed Lagrangian composition FDF code and the Finite Differences method applied to the filtered composition transport equations. Further, the statistical equivalence between the Eulerian and Lagrangian approaches was assessed. We could conclude based on these simulations:

- As the number of notional particles per cell increases, the SDE cases tend to match the PDE result;

- Sorting particles per composition value during annihilation influences SDE results because it tends to keep the composition variance of particle ensemble. Additionally, sorting is more significative than number of particles per cell;

- As pointed out by the literature on mixing flows, the Reynolds number has great influence on species mixing. The laminar ($Re = 50$) and turbulent ($Re = 5000$) flows presented a maximum composition variance of about $10^-4\,(kg/kg)^2$ and $10^-2\,(kg/kg)^2$, respectively;

- The vorticity field changes according to the Reynolds number and influences the mixing pattern, such as position of mixing boundary;

- The results we obtained when comparing the temporal composition average of Partial Differential Equations (PDE) and Stochastic Differential Equations (SDE) simulations showed the Lagrangian FDF algorithm provides solutions statistically equivalent to those obtained using Finite Differences method, even on a complex adaptive, block-structured mesh.

CHAPTER VI

FUTURE WORK

Regarding applications, although we presented only enclosed flow simulations, it is possible to simulate free-shear flows, as well as more complex flows by just changing the initial and the boundary conditions — which we already implemented — for Eulerian variables and notional particles.

Another functionality one can develop and use is the mesh refinement criterion based on composition gradient, additionally to the vorticity criterion. For some flows these criteria will produce meshes quite equals (e.g, mixing shear layer), but they will not for others — for instance, the enclosed mixing flow we simulated. It is specially important for combustion flows, where there are great composition and temperature gradients.

The greatest advantage of a hybrid LES / Lagrangian FDF method is its applicability to turbulent, reacting flows with a large set of chemical-species. Therefore, one can expand the joint AMR3D / Lagrangian FDF code in order to simulate the physics of these flows. It demands the modeling of fluid properties dependent on composition, with different chemical species diffusivities. Although the algorithm already communicates the fluid density field from Lagrangian to Eulerian framework, this field is also influenced by temperature in reaction cases.

One can add the temperature filtered density function to start using combustion models, for instance the simplest "mixed is burnt", Eddy Dissipation Concept. This model requires the transport of a single mixture fraction scalar.

Other potential of the Lagrangian framework is the possibility to implement a joint velocity-composition FDF model based solely on this framework.

REFERENCES

ANSARI, N.; GOLDIN, G. M.; SHEIKHI, M. R. H.; GIVI, P. Filtered density function simulator on unstructured meshes. *Journal of Computational Physics*, v. 230, p. 7132 – 7150, 2011.

BASS, C. A.; BARAT, R. B. Simulation of a toroidal jet-stirred combustor using a partially stirred reactor model with detailed kinetic mechanisms. *Combustion and Flame*, v. 135, p. 249 – 259, 2003.

BOUCHEZ, J.; LAJEUNESSE, E.; GAILLARDET, J.; FRANCE-LANORD, C.; DUTRA-MAIA, P.; MAURICE, L. Turbulent mixing in the amazon river: The isotopic memory of confluences. *Earth and Planetary Science Letters*, v. 290, p. 37 – 43, 2010.

BREIDENTHAL, R. Structure in turbulent mixing layers and wakes using a chemical reaction. *J. Fluid Mech.*, v. 109, p. 1 – 24, 1981.

BROADWELL, J. E.; BREIDENTHAL, R. E. A simple model of mixing and chemical reaction in a turbulent shear layer. *J. Fluid Mech.*, v. 125, p. 397 – 410, 1982.

BUCH, K. A.; DAHM, W. J. A. Experimental study of the fine-scale structure of conserved scalar mixing in turbulent shear flows. part 1. $sc \gg 1$. *J. Fluid Mech.*, v. 317, p. 21 – 71, 1996.

BUCH, K. A.; DAHM, W. J. A. Experimental study of the fine-scale structure of conserved scalar mixing in turbulent shear flows. part 2. $sc \approx 1$. *J. Fluid Mech.*, v. 364, p. 1 – 29, 1998.

COLUCCI, P. J.; JABERI, F. A.; GIVI, P.; POPE, S. B. Filtered density function for large eddy simulation of turbulent reacting flows. *Physics of fluids*, v. 10, n. 2, 1998.

CORREA, S. M.; BRAATEN, M. E. Parallel simulations of partially stirred methane combustion. *Combustion and Flame*, v. 94, p. 469 – 486, 1993.

DUBEY, A.; ANTYPAS, K.; DALEY, C. Parallel algorithms for moving lagrangian data on block structured eulerian meshes. *Parallel Computing*, v. 37, p. 101 – 113, 2011.

FERZIGER, J.; PERIC, M. *Computational methods for fluid dynamics.* [S.l.]: Springer, 1996.

FOX, R. O. *Computational models for turbulent reacting flows.* [S.l.]: Cambridge University Press, 2003.

HANSON, T. D. Uthash. https://github.com/troydhanson/uthash, 2013.

HAWORTH, D. C. Progress in probability density function methods for turbulent reacting flows. *Progress in Energy and Combustion Science*, v. 36, p. 168 – 259, 2010.

KENDALL, W. *Beginning MPI - An introduction in C.* [S.l.: s.n.], 2013.

KONRAD, J. H. *An experimental investigation of mixing in two-dimensional turbulent shear flows with applications to diffusion-limited chemical reactions.* Thesis (Doctorate) — California Institute of Technology, 1977.

KOOCHESFAHANI, M. M.; DIMOTAKIS, P. E. Mixing and chemical reactions in a turbulent liquid mixing layer. *J. Fluid Mech.*, v. 170, p. 83 – 112, 1986.

LIMA, R. S. de. *Desenvolvimento e implementação de malhas adaptativas bloco-estruturadas para computação paralela em mecânica dos fluidos.* Thesis (Doctorate) — Universidade Federal de Uberlândia, 2012.

MELO, R. R. S.; VEDOVOTO, J. M.; VILLAR, M. M.; NETO, A. S. Simulação de escoamentos turbulentos sobre geometrias complexas utilizando malha adaptativa e a metodologia da fronteira imersa. *XXXV Iberian Latin American Congress on Computational Methods in Engineering*, 2014.

METROPOLIS, N. The beginning of the monte carlo method. *Los Alamos Science Special Issue*, 1987.

ORBEGOSO, E. M. M. *Estudo de modelos de mistura estocásticos para a combustão em escoamentos turbulentos.* Dissertation (Master) — Pontifícia Universidade Católica do Rio de Janeiro, Setembro 2007.

PAUL, E. L.; ATIEMO-OBERG, V. A.; KRESTA, S. M. *Handbook of industrial mixing: Science and practice.* [S.l.]: John Wiley & Sons, 2004.

POINSOT, T.; VEYNANTE, D. *Theoretical and Numerical Combustion.* [S.l.]: R. T. Edwards, Inc, 2005.

POPE, P. B. Lagrangian pdf methods for turbulent flows. *Annu. Rev. Fluid Mech.*, v. 26, p. 23 – 63, 1994.

SHAFFER, C. A. *Data structures and algorithm analysis.* [S.l.]: Dover Publications, 2013.

SILVEIRA-NETO, A. *A turbulência nos fluidos aplicada.* [S.l.]: Apostila da disciplina mecânica dos fluidos do programa de pós-graduaçâo da Universidade Federal de Uberlândia, 2002.

SOUZA, F. J. de; OLIVEIRA, J. E. S. Turbulência em motores de combustão interna. *Turbulência*, v. 7, 2010.

VEDOVOTO, J. M. *Mathematical and numerical modeling of turbulent reactive flows using a hybrid LES / PDF methodology.* Thesis (Doctorate) — Universidade Federal de Uberlândia, 2011.

VILLAR, M. M. *Análise numérica fina de escoamentos multifásicos bidimensionais.* Thesis (Doctorate) — Universidade Federal de Uberlândia, 2007.