



Numerical Simulations of Multiphase Flows

ATOMIZATION

Júlia Turró Comas

Supervised by

Stéphane Zaleski

Stanley Yue Ling

Paris, July 9th 2014

ACKNOWLEDGEMENTS

I would like to express my sincere thankfulness to Mr. Stéphane ZALESKI and Mr. Stanley LING for their continued support and assistance, without which I could not have got where I am.

I wish to thank Mr. Gilou AGBADAH for sharing his project with me and being glad to help during the whole internship. In the same way, I would like to expose my gratitude to the other members of the FCIH group, with a special mention to Mr. Daniel FUSTER, Mr. Stéphane VINCENT, Mr. Bertrand LAGRÉE, Mr. Philippe RAY and Mr. Stéphane POPINET. I also want to express a distinctive recognition to Mr. Olivier LABBEY for his assistance with the administrative stuff.

I wish to show recognition for Mr. Grétar TRYGGVASON, who suggested one of the cases.

I also want to express my gratitude to the CINES engineers who helped to install the software Gerris on Jade.

At last but not least, I extend my gratefulness to all my intern partners in the d'Alembert during these five months, particularly to my colleague Rafael SÁNCHEZ.

CONTENTS

1. INTRODUCTION	7
1.1. Motivation of study	7
1.2. Study objective	7
1.3 Internship environment.....	8
1.3.1. Jean le Rond d'Alembert Institut.....	8
1.3.2. Complex Fluids and Hydrodynamic Instabilities.....	8
2. COMPUTATIONAL FLUID DYNAMICS	10
2.1. Introduction.....	10
2.1.1. Discretization methods.....	11
2.1.2. Turbulence simulation approaches	12
2.2. Direct numerical simulation	13
2.3. Volume of fluid method	14
2.4. Other methods	15
2.4.1. Front Tracking.....	15
2.4.2. Level-set method.....	16
3. ATOMIZATION	17
3.1. The phenomenon	17
3.2. Applications.....	18
4. EQUATIONS AND NUMERICAL METHOD	19
4.1. Governing equations	19
4.2. Numerical Method: Gerris.....	20
4.3. Machine for the simulations	21
5. SIMULATIONS CHARACTERIZATION	22
5.1. Problem description	22
5.2. Simulations set-up.....	22
5.2.1. Case A.....	24
5.2.2. Case B	26
5.2.3. Case C	27
6. RESULTS	29
6.1. Case A.....	29
6.1.1. Case A at level 9.....	29
6.1.2. Case A at level 10	31
6.2. Case B	35

6.3. Case C	37
6.4. Comparison	38
6.5. Gerris performance	40
6.5.1. Real Solid vs. Virtual Solid	40
6.5.2. Jade 1 vs. Jade 2	40
CONCLUSIONS AND NEXT STEPS	42
BIBLIOGRAPHY.....	43

1. INTRODUCTION

1.1. Motivation of study

Nowadays, numerical simulations are becoming more and more relevant in the scientific and technological world. They prevent wrong designs, give ideas of better solutions and lead a greater understanding of new technology operations. Hence, in a very large number of industries, before manufacturing a product many different simulations are need to be made; and this fact even increases when we talk about technological stuff. So, from the point of view of an industrial engineering student, numerical simulations weighs heavily since there is a high probability to encounter them in the professional life.

On the other hand, many of the computational simulations mentioned above have to do with fluid dynamics, the sub-discipline of fluid mechanics that deals with fluid flows. Some applications of fluid dynamics may be aeronautics, aeroacoustics, environmental engineering, biomechanics and road vehicle aerodynamics. Consequently, there are many aspects of the numerical simulations (how far it can reaches, its shortages, etc.) that need to be profoundly studied.

A concrete phenomenon related to fluid dynamics is atomization. When talking about atomization or an atomizer we may refer to the making of an aerosol, any kind of spray, a nebulizer, a component in electronic cigarettes, an atomic spectroscopy or even to fogs and clouds which appear to be atomized. So, it remains clear that the phenomenon of atomization is present in a great amount of different processes. Therefore, it needs to be deeply studied seeing that, although a lot has been made, there are still many aspects to be resolved.

1.2. Study objective

The objective of the study presented in this report is mainly to understand the behavior of multiphase flows. As many studies on this field have already done, the main aim is to get further resolved results on the phenomenon of atomization.

On the other hand, another goal is to analyse the many different options to run the codes in order to decide which alternative turns out in the most optimized result.

1.3 Internship environment

1.3.1. Jean le Rond d'Alembert Institut

The Jean le Rond d'Alembert Institut (d'Alembert from now on) is a research laboratory whose mission is to extend the knowledge in all the domains of Mechanics, Acoustics and Energy. The d'Alembert is a mixed unit of research between the Université Pierre et Marie Curie (Paris VI) and the Centre Nationale de la Recherche Scientifique (CNRS), and it is also supported by the Ministry of Culture. It is the greatest research laboratory of the region Île-de-France on its domain. It gathers nearly one hundred and seventy persons: a hundred of permanents, around fifty doctorate students and about twenty temporary employees, post docs and others.

The d'Alembert has a unique expertise in two concrete fields: the theory and the fine modeling in fluid and solid mechanics, and the study of the musical object in a multidisciplinary approach, through the physical and human sciences. The researchers from the d'Alembert rank first in the world in the domain of fracture mechanics, direct numerical simulation of multiphase flows, study of slender structures, simulation of large turbulent structures (Large Eddy Simulation), non-linear acoustic propagation and of sonic boom. At the same time, new themes such as aeroacoustics (sound generation by turbulence) or dynamic elastocapilarity (winding elastic solid structures around liquid droplets) emerge through cross-collaborations. The d'Alembert also implements leading experiments supported by large industrial and public partnerships in the area of combustion engines, as well as acoustic imaging. [5]

The institute is divided into five different research groups: Complex fluids and hydrodynamic instabilities (FCIH); Reactive fluids and turbulence (FRT); Lutheries – Acoustics – Music (LAM); Mechanics and engineering of solids and structures (MISES); Modeling, propagation and acoustics imaging (MPIA).

1.3.2. Complex Fluids and Hydrodynamic Instabilities

Nowadays fluid mechanics deals with problems of ever increasing complexness. Complexity may arise because the Navier-Stokes equation, though deceptively simple instigates ever more complex solutions such as contorted vortical flows. Hydrodynamic instabilities take place in a bewildering variety of symmetries and mechanisms. The entire complex phenomenology of droplets and bubbles is acquired when interfaces and capillarity are added. In atomizing jets and sprays or sedimenting flows, an additional layer of complexity can be created by a considerable number of droplets and bubbles or solid particles. In the end, living systems present really astonishing complexity. Hence, the group's core interests are flows in the nature

and environmental fluid mechanics.

Usually, a great part of the research in the group is “curiosity driven”. Nevertheless, countless sections of target-oriented research are based on the fluid mechanics of complex flows. A common theme in the group is to treat the resulting complexity as a bona fide scientific inquiry, in line with the currently hot topics in the international community of fluid mechanics, theoretical physics and applied mathematics. [5]

2. COMPUTATIONAL FLUID DYNAMICS

2.1. Introduction

Computational fluid dynamics (CFD) is a section of fluid mechanics that combines numerical methods, mathematical modeling and software tools to solve and analyze problems related to fluid flows. The interaction of liquids and gases with surfaces, defined by boundary conditions, can be simulated using modern computers to perform the calculations.

CFD allows the fluid mass flow, velocity, density, temperature, pressure, concentration or any derived fluid property to be calculated at all locations within the flow field. The interaction of the fluid with its solid boundaries, such as fluid-dynamic forces and heat transfer rates can also be calculated accurately for use in stress analysis. The fundamental basis of almost all CFD problems are the Navier–Stokes equations, which define any single-phase (gas or liquid, but not both) fluid flow. [12]

The fidelity and accuracy of modern CFD methods has notably increased the level of design insight available to engineers throughout the design process and therefore reduces significantly the exposure of companies to technical risk when developing fluid and thermal-based products. By using this technique, designers can verify that their products will conform to a client’s specifications early in the design cycle, accelerating the product development process. Usually, the usage of CFD in design leads to far less physical prototypes being necessary during development, far fewer prototype testing and consequently reduces the time-to-market and cost-to-market considerably.

Applications of CFD include aerodynamics, industrial fluid dynamics, fluid structure interaction, heat transfer, hydrodynamics and multi-phase flows, among others.

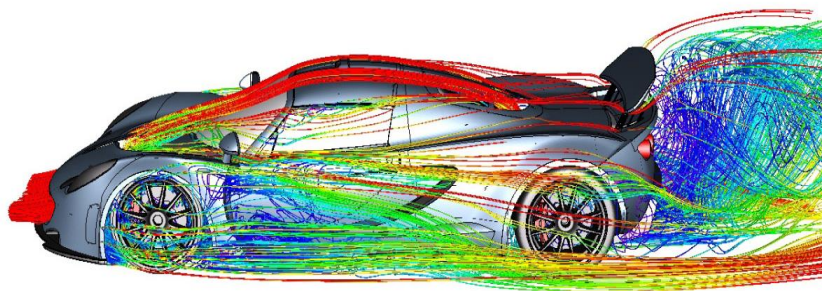


Figure 1 CFD applied to a car aerodynamics

All the different approaches follow the same procedure. During the preprocessing, the geometry (physical boundary) of the problem is defined, the domain is divided into discrete cells (the mesh), and the physical modeling is defined, as well as the boundary and the initial conditions. Then, the simulation is started and the equations are solved iteratively. Finally, a postprocessor is used for the analysis and visualization of the results. [12]

2.1.1. Discretization methods

Generally, and on the contrary to simple linear problems, no analytical solution exists for non-linear models. The stability of the selected discretization is normally established numerically. Some of the most commonly-used discretization methods are the ones that follow.

- *Finite element method:*

The finite element method (FEM) is used in structural analysis of solids and fluids. Although its formulation requires special care to be conservative, it uses vibrational methods (the calculus of variations) to minimize the error function and produce a veritably stable approach.

FEM includes all the methods to connect various simple element equations over many little subdomains to approximate a more complex equation among a larger domain.

- *Finite volume method:*

Similarly to the FEM, the finite volume method (FVM) calculates the values at discrete places on a meshed geometry, but the governing partial differential equations are solved over discrete control volumes surrounding each node point. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative. This discretization guarantees the conservation of fluxes through a particular control volume.

Although FVM requires less memory and assures to be conservative, it has higher solution times and is less stable than the FEM approach.

- *Finite difference method:*

The finite difference method (FDM) approximates the solutions to differential equations using finite difference equations to approximate derivatives. They are used for problems which handle complex geometries with high accuracy and efficiency.

2.1.2. Turbulence simulation approaches

Computational modeling of turbulent flows usually involves a range of length scales and complexity that contributes to approaches prohibitively expensive, since the resolution required is beyond what is computationally possible. The primary approach in such cases is to create numerical models to approximate unresolved phenomena.

Turbulence models can be classified according to the range of scales that are modeled versus resolved, that is, computational expense. Although the computational cost is very low if most of turbulence scales are modeled, this also leads to worse accuracy. In addition, the governing equations of fluid-dynamics contain a non-linear convection term and a non-linear and non-local pressure gradient term which have to be solved numerically with the proper initial and boundary conditions. [13]

Some of the most commonly-used turbulence approaches are listed below.

- Direct numerical simulations:

Direct numerical simulation (DNS) model resolves the entire range of turbulent length scales, which means a significant computational cost. Since this technique is the turbulence model used for the simulations presented in this report, it will be widely detailed in the next section.

- Reynolds-averaged Navier-Stokes:

Reynolds-averaged Navier-Stokes equations (RANS) were the first approach to turbulence modeling. They are based on the idea that an instantaneous quantity is decomposed onto its time-averaged and fluctuating quantities¹. The knowledge of the properties of flow turbulence eases to use these equations with approximations to give estimated time-average solutions to the Navier-Stokes equations.

- Large eddy simulation:

Large eddy simulations (LES) is a technique based on low-pass filtering applied to the Navier-Stokes equations to eliminate small scales of the solution reducing the computational cost of the simulation. The selection of which of the small length and time scales have to be eliminated is done according to turbulence theory and available computational resources. [13]

¹ This idea is a mathematical technique called Reynolds decomposition, and it was first proposed by Osborne Reynolds.

Although resolving large scales of the flow field solutions allows LES a better fidelity than RANS, it requires greater computational resources. On the other hand, it is far cheaper than DNS.

2.2. Direct numerical simulation

Direct Numerical Simulation (DNS) is the branch of CFD devoted to high-fidelity solution of turbulent flows. DNS differs from conventional CFD in that the turbulence is explicitly resolved, rather than modelled by a Reynolds-Averaged Navier-Stokes (RANS) closure. It differs from Large-Eddy Simulation (LES) in that all scales, including the very smallest ones, are captured, removing the need for a subgrid-scale model. DNS can thus be viewed as a numerical experiment producing a series of non-empirical solutions, from first principles, for a virtual turbulent flow. Its great strength is the ability to provide complete knowledge, unaffected by approximations, at all points within the flow and at all times within the simulation period. DNS is therefore ideal for addressing basic research questions regarding turbulence physics and modelling.

The defining characteristics of DNS stem from the distinctive characteristics of turbulence. Because turbulence is inherently unsteady and three-dimensional, DNS requires time-dependent calculation within a three-dimensional domain. These two features are shared with LES (and therefore LES/RANS hybrid strategies such as detached eddy simulation (DES)). The unique feature of DNS is associated with the manner in which turbulence is affected by viscosity. This is responsible for the two chief drawbacks of DNS – its extreme computational cost, and severe limitation on the maximum Reynolds number than can be considered.²

DNS has become a standard tool in turbulence research of homogeneous flows and notable results have been accomplished. However, and unsurprisingly, simulations of multiphase flows have remained far behind, not due to lack of effort though. Actually, in the last decade and a half or so, these efforts have started to be worth it and rather significant progress has been achieved on many fronts. It is now possible to do DNS for a great number of fairly complex systems and DNS are starting to yield information that is likely to be unobtainable in any other way.

² Coleman, G. N., and Sandberg, R. D. *A primer on Direct Numerical Simulation of Turbulence – Methods, Procedures and Guidelines*. ePrints Soton – University of Southampton (School of Engineering Sciences), March 2010, [http://eprints.soton.ac.uk/66182/1/A_primer_on_DNS.pdf , July 16th 2014].

2.3. Volume of fluid method

The volume of fluid (VOF) method is a numerical technique historically used for tracking and locating free surface flows with only one fluid, although it is now regularly used for two-fluid flows. It belongs to the class of Eulerian methods which are characterized by a mesh that is either stationary or is moving in a certain prescribed manner to accommodate the evolving shape of the interface. In the VOF method the marker function is represented by a fraction of computational grid cell which is occupied by the fluid assumed to be the reference phase.

The method is based on the so-called volume fraction or color fraction function C . This function varies between the constant value one in full cells to zero in empty cells, while mixed cells with an intermediate value of C define the transition region where the interface is localized. It is defined as the average value of H^3 in each computational cell. In other words, C is the discrete functions of H . So, for a rectangular two-dimensional cell

$$C_{i,j} = \frac{1}{\Delta x \Delta y} \int_V H(x,y) dx dy.$$

Eq. 1

A VOF method is performed in two steps:

- 1.- Reconstruction of an approximation of the interface shape by knowing the volume fraction in each cell.
- 2.- Advection of the reconstructed interface in a given velocity field.

Lower-order VOF methods such as the Simple Line Interface Construction (SLIC) reconstruction do not need to specify the location of the interface in the transition region, although geometrical interpretations of these methods show that in two dimensions the interface line in each mixed cell is represented by a segment parallel to one coordinate axis. Hence, the interface is definitely not continuous across the cell boundary. Regarding higher-order methods, such as the standard Piecewise Linear Interface Construction (PLIC) reconstruction, the interface can be reconstructed in various ways. Despite the fact that the interface at the cell boundary is not continuous yet, the interface discontinuity is normally much smaller and usually it is a function of the grid spacing h and of the local interface curvature κ .⁴

³ The H function is used to identify whether a given fluid is present at a particular location x . It is defined by

$$H_i(x) = \begin{cases} 1, & \text{if } x \text{ is in fluid } i; \\ 0, & \text{if } x \text{ is not in fluid } i. \end{cases}$$

Eq. 2

⁴ Tryggvason, G., Scardovelli, R. and Zaleski, S. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge (2011): Cambridge University Press, 95.

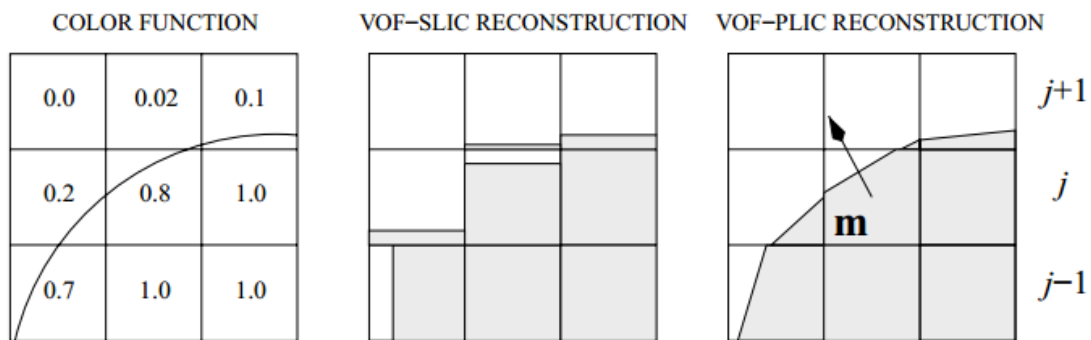


Figure 2⁵ The basic principle of the VOF methods: color function value in each cell, SLIC reconstruction, and PLIC reconstruction; the normal vector m is pointing outwards from the reference face.

In PLIC methods, after the reconstruction, the interface is advected in a given velocity field up to the next discrete time. These evolutions may be estimated with geometrical approaches, by advecting the end points of each segment of the interface, or by fluxing approaches, by computing the reference phase fluxes across the cell boundary. [11]

2.4. Other methods

Although VOF is the most significant method for the present report since it is the one used for the simulations done, there are other relevant methods for advecting a fluid interface that also deserve a brief presentation.

2.4.1. Front Tracking

Contrary to the VOF method, which advects a marker function by reconstructing the interface identifying the different fluids directly, the boundary between the fluids can also be represented by connected marker points that are moved with the imposed velocity. Usually, to track the boundary between different fluids or phases with connected marker points it is necessary to decide how the front is managed and represented as it stretches and deforms. It is also required to know how the interface interacts with the underlying grid used to solve the equations governing the fluid flow, how the interface is advanced in time, and how changes the topology of the interface when fluid blobs merge and break apart.

Since most of front-tracking methods use triangulated unstructured grids to represent the interface, grid points can be added or deleted as it stretches or compresses. For multiphase flow simulations, the effort needed to manage the front is usually small compared with the overall effort required for the simulation. [11]

⁵ Tryggvason, G., Scardovelli, R. and Zaleski, S. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge (2011): Cambridge University Press, 96.

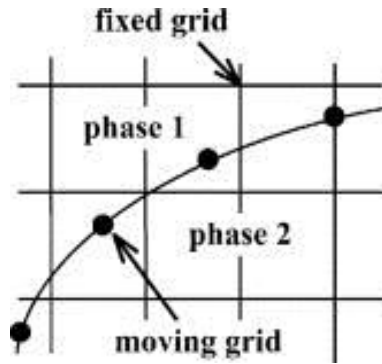


Figure 3 Simplified diagram of the front tracking method for the advection of the surface

2.4.2. Level-set method

In this method for advecting a fluid interface the identification of the different fluid regions is done by a smooth marker function $F(x, t)$. This level-set function is positive in one fluid and negative in the other, thus the interface between the fluids is located by the level curve $F(x, t) = 0$.

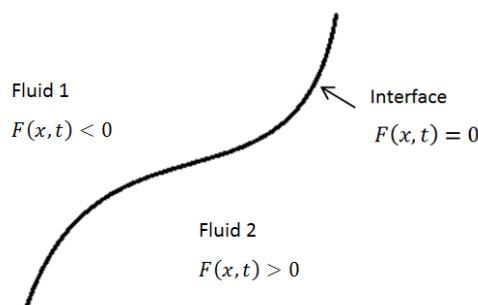


Figure 4 Simple diagram of the level-set function

The level-set method has the appeal of simplicity; one can perform computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects. In addition, while the VOF and the front tracking methods need additional steps after solving the additional partial differential equation (the reconstruction of the interface and the addition of new computational objects, respectively), in the level-set method no additional complex step is needed. However, this simplicity turns out into complexity since mass conservation is not assured, and more accurate methods are required. [8]

3. ATOMIZATION

3.1. The phenomenon

Atomization is a phenomenon where a liquid jet is broken into seemingly microscopic droplets, although the term is somewhat incorrect, since the individual pieces are still far larger than atomic scales. Nevertheless, it is a striking process in which finely divided sprays or droplet clouds are produced. This is often based on the ejection of a high-speed liquid jet from an atomizer nozzle. Many other configurations exist, such as sheets ejected at high speed from diversely shaped nozzles, or colliding with each other. [11]

One of the most difficult conceptual issues in atomization is to understand the physical mechanisms that lead to atomization. Several such mechanisms have been proposed and confirmed experimentally. The three most important ones are the effect of upstream liquid-turbulence, the stability of idealized liquid jets or sheets, and cavitation in the liquid. [6, 11]

The liquid-turbulence theory explains droplet formation as a consequence of the deformation of the interface by strong vortices in the liquid, which requires Re_l (liquid Reynolds number) to be sufficiently large for turbulence to be well developed in the liquid. The stability theory instead assumes an idealized flow out of the nozzle. Finally, cavitation mechanisms arise when regions of very low pressure are present upstream of the nozzle, leading to the growth of vapor bubbles or sheets in the liquid.

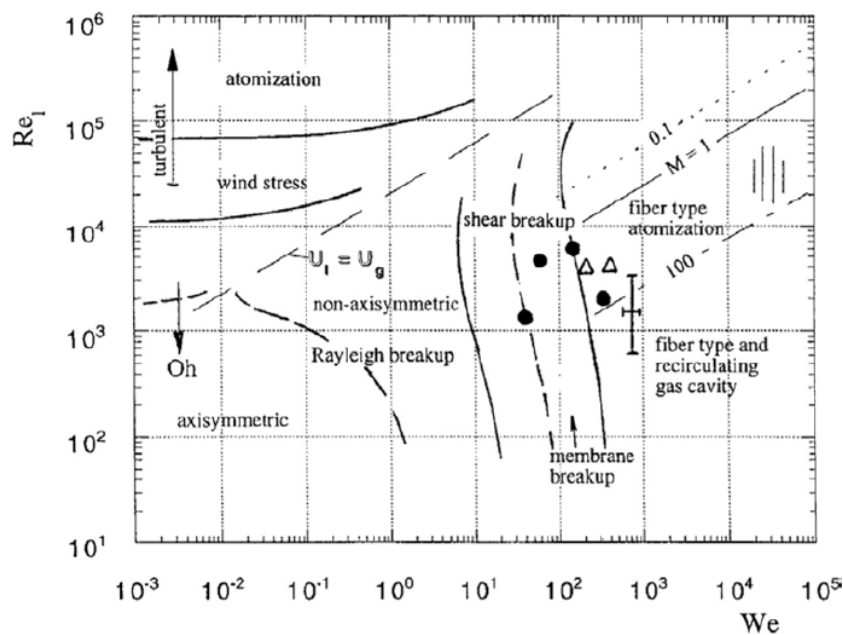


Figure 5 Hopfinger diagram of the various atomization regimes as a function of the liquid Reynolds number Re_l and the gas Weber number We_g .

It is reasonable to assume that the various theories apply in different regions of parameter space. High liquid Reynolds number would lead to liquid-turbulence effects, while high gas Weber number leads to aerodynamic effects (look at Figure 5).⁶

3.2. Applications

The process of atomization is of importance since various industrial and natural processes involve great deformations and eventual breakup of liquid jets, layers and droplets.

Concerning the industrial phenomena, spray properties need to be studied and controlled by engineers in several ways. In some cases it is useful to have large droplets as they create mixing, while in other cases relatively narrow distributions of droplet sizes are required to ensure uniform evaporation of the droplets. [11] Another especially relevant application is combustion, since a great amount of well-known industrial processes are based on this exothermic chemical reaction. For combustion to occur, the liquid fuel has to be atomized, evaporated and mixed with air. Atomization is the initial step in the energy conversion process and thus greatly influences all subsequent processes.

From the natural point of view, a first example is spray formation atop the ocean waves when sufficiently strong winds takes away droplets from the top of the waves. Breaking waves also origins bubbles that, when rupturing at the surface, create a very fine mist that can raise high into the atmosphere. These different phenomena related to ocean waves play a significant role in ocean-atmosphere exchanges and interactions, which is directly associated with climate-change dynamics.⁶

Consequently, due to the large number of processes governed by the phenomenon of atomization, it has become of great importance to study and understand its behaviour. Numerical simulations of this process may emerge significant results which could mean great progress in some industries.

⁶ Tryggvason, G., Scardovelli, R. and Zaleski, S. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge (2011): Cambridge University Press, 215-216/204.

4. EQUATIONS AND NUMERICAL METHOD

4.1. Governing equations

Fluid flows are governed by the so called Navier-Stokes equations. For incompressible and variable-density, Navier–Stokes equations with surface tension can be written

$$\rho(\delta_t u + u \cdot \nabla u = -\nabla p + \nabla \cdot (2\mu D) + \sigma k \delta_s n,$$

$$\delta_t \rho + \nabla \cdot (\rho u) = 0,$$

$$\nabla \cdot u = 0,$$

Eq. 2, 3, 4

with $u = (u, v, w)$ the fluid velocity, $\rho \equiv \rho(x, t)$ the fluid density, $\mu \equiv \mu(x, t)$ the dynamic viscosity and D the deformation tensor defined as $D_{ij} \equiv (\delta_i u_j + \delta_j u_i)/2$. The Dirac distribution function δ_s expresses the fact that the surface tension term is concentrated on the interface; σ is the surface tension coefficient, k and n the curvature and normal to the interface.⁷

For two-phase flows we introduce the volume fraction $c(x, t)$ of the first fluid and define the density and viscosity as

$$\rho(\tilde{c}) \equiv \tilde{c} \rho_1 + (1 - \tilde{c}) \rho_2,$$

$$\mu(\tilde{c}) \equiv \tilde{c} \mu_1 + (1 - \tilde{c}) \mu_2,$$

Eq. 5, 6

with ρ_1, ρ_2 and μ_1, μ_2 the densities and viscosities of the first and second fluids respectively.

Field \tilde{c} is either identical to c or is constructed by applying a smoothing spatial filter to c .

The advection equation for the density can then be replaced with an equivalent advection equation for the volume fraction⁷

$$\delta_t c + \nabla \cdot (cu) = 0.$$

Eq. 7

⁷ Popinet, S. *An accurate adaptive solver for surface-tension-driven interfacial flows*. Journal of Computational Physics 228 (2009), 5838-5866.

4.2. Numerical Method: Gerris

All the simulations have been carried out by the software Gerris, since it was thought that it would be the best software to get further resolved results, as well as not so difficult to learn as other ones. The choice was also promoted by the fact that Gerris was created by a person based at the Institute where this internship has been done, and that the supervisor also collaborated with its conception.

Gerris is a free open-source software in the field of CFD available free of charge under the Free Software GPL license. Created by Stéphane Popinet⁸ and supported by NIWA⁹ and d'Alembert, it resolves the partial differential equations describing fluid flows.

Gerris solves the time-dependent incompressible variable-density of Euler, Stokes and Navier-Stokes equations, as well as linear and non-linear shallow-water equations. On the contrary, it does not allow the modeling of compressible fluids. While most models use meshes which are either structured or unstructured, Gerris implements a deal between structured and unstructured meshes by using a tree data structure. This leads to an adaptive mesh refinement where the resolution is adapted dynamically to the features of the flow. Moreover, it does not need a meshing tool since the refinement of the grid is on charge of the solver itself. Even in complex geometries the mesh generation is completely automatic. [4]

The advection scheme used for interfacial flows is the Volume of Fluid, and the number of advected/diffused passive tracers is unlimited. Gerris mainly aims at DNS; the range of Reynolds available to the user thus depends on the computing power he/she can afford (although the auto-adaptive mesh allows one to focus the computing resources on the coherent structures). On the other hand, it utilizes a very accurate surface tension model. [4]

The combination of a quad/octree discretisation, balanced-force CSF surface tension scheme and generalised height-function curvature estimation implemented within Gerris has been demonstrated to give accurate and efficient solutions for surface-tension-driven flows. The VOF method, height-function and CSF formulations have been generalised to a fully-adaptive quad/octree discretisation allowing refinement along the interface. For the case of capillary breakup of a three-dimensional liquid jet, this leads to a reduction by a factor of fifty of the mesh size compared to using a constant resolution along the interface. A staggered in time discretization of the volume-fraction/density and pressure leads to the following formally

⁸ Director of research at the CNRS based at the UPMC.

⁹ National Institute of Water and Atmospheric research.

second-order accurate time discretization¹⁰

$$\rho_n + \frac{1}{2} \left[\frac{u_{n+1} - u_n}{\Delta t} + u_{n+\frac{1}{2}} \cdot \nabla u_{n+\frac{1}{2}} \right] = -\nabla p_{n+\frac{1}{2}} + \nabla \cdot \left[\mu_{n+\frac{1}{2}} (D_n + D_{n+1}) \right] + (\sigma k \delta_s n)_{n+\frac{1}{2}},$$

$$\frac{C_{n+\frac{1}{2}} - C_{n-\frac{1}{2}}}{\Delta t} + \nabla \cdot (c_n u_n) = 0,$$

$$\nabla \cdot u_n = 0.$$

Eq. 8, 9, 10

Gerris is developed in C using the libraries Glib (object orientation, dynamics loading of modules...) and GTS (3D surfaces meshed with interconnected triangles). In addition, Gerris includes portable parallel support using the MPI library, as well as dynamics load-balancing and parallel offline visualization.

4.3. Machine for the simulations

All the following simulations were performed on the supercomputer Jade (SGI) based at the French National Computer Center for Higher Education (CINES) established in Montpellier. Jade machine was ranked 27th among the world's computers TOP500 on November 2010, it is the 6th European machine and the first French machine for public research.

The cluster SGI Altix ICE 8200 – JADE is a scalar parallel supercomputer of a maximum power of 267 Tflop/s. It consists of 46 racks: 45 racks of calculation and one which provides the connection to the entire cluster. This cluster includes 23040 cores shared out over 2880 nodes (each one with two Intel Quad-Core E5472 and X5560 processors). Each node has 30 GB of useful memory, so there are more than 91 TB in total. Racks for calculations are connected to 12 racks mounted on a shared system with a capacity of 689 TB useful in total. The cooling process is provided by a high efficiency air/water system. [1]

¹⁰ Popinet, S. *An accurate adaptative solver for surface-tension-driven interfacial flows*. Journal of Computational Physics 228 (2009), 5838-5866.

5. SIMULATIONS CHARACTERIZATION

5.1. Problem description

The cases presented are mainly based on two different papers: *Break-up and atomization of a round water jet by a high-speed annular air jet* by J. C. Lasheras, E. Villermaux and E. J. Hopfinger on 1998; and *On spray formation* by P. Marmottant and E. Villermaux on 2004.

As mentioned before in this report, the dispersion and disintegration of a liquid volume by a gas stream is a process which encompasses many natural and industrial operations. In order to estimate the size of a combustion chamber, or to compute the rate of exchanges between the ocean and the atmosphere, it is usually desirable to have a precise knowledge of the liquid dispersion structure and its distribution of droplet sizes as a function of the external parameters (liquid surface tension, air speed, etc.). So, this is the reason why these simulations were carried out.

The very basic idea of the simulations was to represent a nozzle which injects liquid water at high pressure and is wrapped by air. The set up consists of two coaxial jets: the one situated in the interior is the one that injects the liquid, and the one outside inserts gas at a greatly higher velocity (look at Figure 6).

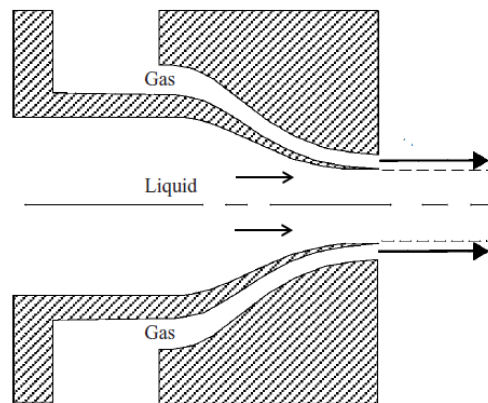


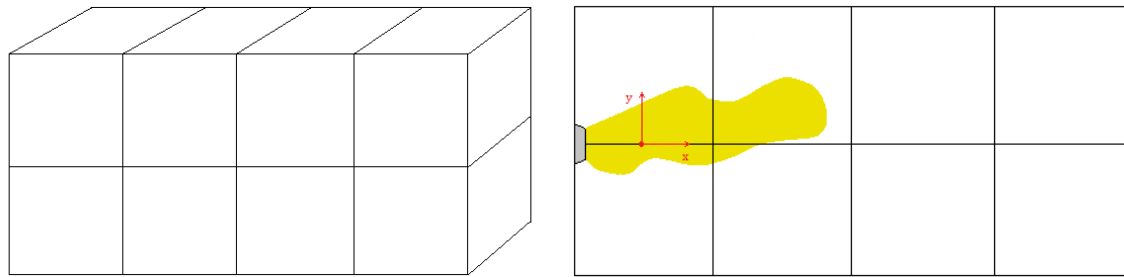
Figure 6¹¹ Simplified diagram of the injector

5.2. Simulations set-up

In the current paper three different cases are presented: case A, case B and case C. Case A and case B are both based on G. Agbalah's simulations, and their parameters almost only differ on the dynamic viscosities. On the other hand, case C is sustained on the paper by J. C. Lasheras *et al.* (1998).

¹¹ Marmottant, P., and Villermaux, E. *On spray formation*. Journal of Fluid Mechanics 498 (2004), 76.

The original file for all the three cases included 8 boxes arranged as in the following diagrams:



(a)

(b)

Figure 7 Simplified diagrams: (a) disposition of the boxes, (b) nozzle, liquid interface and domain.

The parameters defined in both three cases are the following:

Simbol	Parameter	Simbol	Parameter
ρ_l	Liquid density	r	Liquid jet radius
ρ_g	Gas density	a	Separation plate thickness
μ_l	Liquid dynamic viscosity	h	Thickness of gas layer
μ_g	Gas dynamic viscosity	<i>length</i>	Nozzle length
U_l	Liquid velocity	<i>level</i>	Level
U_g	Gas velocity	δ_g	Gas boundary layer thickness
σ	Surface tension	l	Box length

Table 1 Parameters that will be defined in each case and their terminology

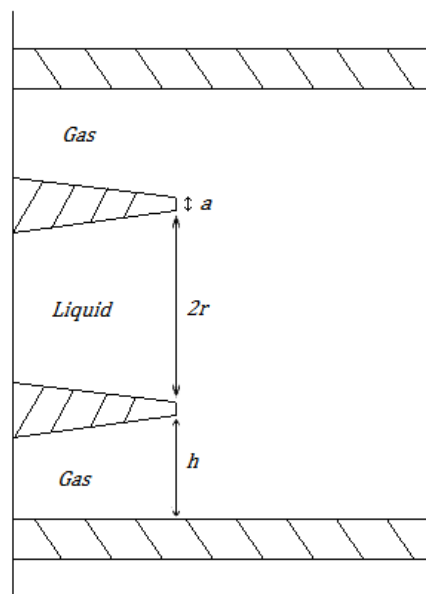


Figure 8 Simplified diagram of the solid measurements

Using the parameters defined in Table 1 many dimensionless numbers are calculated for each case:

Dimensionless parameter	Terminology and formula
Momentum flux ratio per unit volume	$M = \frac{\rho_g U_g^2}{\rho_l U_l^2} \quad \text{Eq. 11}$
Gas Webber number referred to h	$We_{gas,h} = \frac{\rho_g U_g^2 h}{\sigma} \quad \text{Eq. 12}$
Ratio between the aerodynamic deformation pressure force exerted on the liquid and the restoring surface tension forces	$We_0 = \frac{\rho_g (U_l - U_g)^2 D_l}{\sigma}, \quad D_l = 2r \quad \text{Eq. 13, 14}$
Gas Webber number referred to the diameter	$We_{gas,D} = \frac{\rho_g U_g^2 (D_l + 2t)}{\sigma} \quad \text{Eq. 15}$
Gas Webber number referred to δ_g	$We_{gas,\delta_g} = \frac{\rho_g U_g^2 \delta_g}{\sigma} \quad \text{Eq. 16}$
Gas Reynolds number referred to D_g	$Re_{gas,D_g} = \frac{\rho_g U_g D_g}{\mu_g}, \quad D_g = 2(r + t + h) \quad \text{Eq. 17, 18}$
Liquid Reynolds number referred to the diameter (D_l)	$Re_{liq,D_l} = \frac{\rho_l U_l D_l}{\mu_l} \quad \text{Eq. 19}$
Gas Reynolds number referred to δ_g	$Re_{gas,\delta_g} = \frac{\rho_g U_g \delta_g}{\mu_g} \quad \text{Eq. 20}$
Gas Reynolds number referred to h	$Re_{gas,h} = \frac{\rho_g U_g h}{\mu_g} \quad \text{Eq. 21}$
Grid size	$\delta_x = \frac{l}{\sqrt{\text{level}}} \quad \text{Eq. 22}$

Table 2 Dimensionless numbers that will be defined and their formulas

5.2.1. Case A

The idea of simulating this case came from the goodwill to get further resolved results than the ones that had been obtained in previous similar studies. In this case the liquid injected by the nozzle is a viscous fluid¹², and it was intended to get something similar to the Figure 9 experiment.

¹² Compared to liquid water.



Figure 9¹³ Breakup regime of a slow dense liquid jet by a fast light coaxial stream for higher gas velocity

The parameters used for case A are the ones listed below.

Parameter	Value	Parameter	Value
ρ_l	1	r	0.1
ρ_g	0.001035	t	0.011634
μ_l	$5.16 \cdot 10^{-4}$	h	0.0436
μ_g	$7.15 \cdot 10^{-7}$	$length$	0.06977
U_l	1	$level$	9/10
U_g	58.3	δ_g	0.004026
σ	$4.3 \cdot 10^{-3}$	l	1

Table 3 Value of the parameters for the case A¹⁴

Using these values the dimensionless numbers for case A are:

Dimensionless parameter	Value	Dimensionless parameter	Value
M	3.516	Re_{gas,D_g}	26982.106
$We_{gas,h}$	35.652	Re_{liq,D_l}	387.597
We_0	157.978	Re_{gas,δ_g}	339.606
$We_{gas,D}$	190.263	$Re_{gas,h}$	3677.691
We_{gas,δ_g}	3.292	δ_x	0.00195/0.0009

Table 4 Dimensionless numbers values for case A

Before submitting the simulation, this case was split into 4096 boxes (since the original file

¹³ Marmottant, P., and Villermaux, E. *On spray formation*. Journal of Fluid Mechanics 498 (2004), 79.

¹⁴ All the values in the whole report are not in the International units system but in Gerris units. They have all been transferred taking as reference the box length to 1.

included 8 boxes, that is each box suffered 3 separations into 8 boxes); and then partitioned into 8, 64, and 512 processors. The partition was done in three different ways in order to compare the results at the beginning stage, and then continue the simulation only with the more efficient one.

5.2.2. Case B

The current case is actually the same as case A except for the liquid, which in this case turns out to be water. It is easy to see that most of the following parameters are the same that in case A, since both cases are based on the same study.

Parameter	Value	Parameter	Value
ρ_l	1	r	0.1
ρ_g	0.0012	a	0.011634
μ_l	$4.87 \cdot 10^{-5}$	h	0.0436
μ_g	$8.29 \cdot 10^{-7}$	<i>length</i>	0.06977
U_l	1	<i>level</i>	9/10
U_g	58.3	δ_g	0.004026
σ	$4.91 \cdot 10^{-3}$	l	1

Table 5 Parameters values for the case B

These parameters lead the next values for the dimensionless numbers:

Dimensionless parameter	Value	Dimensionless parameter	Value
M	4.079	Re_{gas,D_g}	26995.047
$We_{gas,h}$	36.218	Re_{liq,D_l}	4777.823
We_0	166.138	Re_{gas,δ_g}	339.688
$We_{gas,D}$	193.284	$Re_{gas,h}$	3679.455
We_{gas,δ_g}	3.344	δ_x	0.00195/0.0009

Table 6 Dimensionless numbers for the case B

In this case, the job was also split into 4096 boxes in order to get a good balance. However, this simulation was only partitioned into 64 processors.

5.2.3. Case C

This case was carried out due to the desire of acquiring fully resolution on atomization in the early stages. To obtain this, the M number must not be so high, since then there won't be many grid points, and the result will be totally resolved in less time. So to build this case it was took as reference a study made by J. C. Lasheras *et al.* (1998) which's specifications were: $M = 7.35$, $We_0 = 31$ and $Re_g = 1120$.



Figure 10¹⁵ Instantaneous flow visualization of the brak-up of a liquid jet by an annular air jet.

Find below the parameters for this case:

Parameter	Value	Parameter	Value
ρ_l	1	r	0.22
ρ_g	0.05	a	0.022496
μ_l	$1.73 \cdot 10^{-2}$	h	0.0921
μ_g	$8.66 \cdot 10^{-2}$	<i>lenght</i>	0.24
U_l	43.42	<i>level</i>	7/8
U_g	526.32	δ_g	0.00526
σ	182	l	1

Table 7 Parameters values for case C.

These data leads to the following dimensionless numbers, which do not differ much from J. C. Lasheras *et al.* case.

Dimensionless parameter	Value	Dimensionless parameter	Value
M	7.346	Re_{gas,D_g}	2034.376
$We_{gas,h}$	6.999	Re_{liq,D_l}	1103.52

¹⁵ Lasheras, J. C., Villermaux, E., and Hopfinger, E. J. *Break-up and atomization of a round water jet by a high-speed annular air jet.* Journal of Fluid Mechanics 357 (1998), 351-379.

We_0	28.150	Re_{gas,δ_g}	160
$We_{gas,D}$	36.859	$Re_{gas,h}$	280
We_{gas,δ_g}	3.999	δ_x	0.0078/0.0039

Table 8 Dimensionless numbers for the case C.

On the contrary to Case A and B, this case was split into 512 boxes, and the job was run on 64 cores.

On the contrary to case A and B, in this case the nozzle was represented by a virtual solid. For more information look at section 6.5..

6. RESULTS

6.1. Case A

6.1.1. Case A at level 9

As it has been mentioned before, the original file was split three times achieving 4096 boxes. This procedure was carried out by Gerris on the laboratory workstation, and then, after partition it in three different manners, those three files where uploaded on Jade.

Before proceeding to do more changes on the script, the files were submitted on Jade to check whether they run or not. So, as in the original simulations, they were executed with a maximum level of 9. Both the three cases were simulated during 24 hours to compare the results and decide with which to continue. After those 24 hours, depending on the number of processors used, the simulation reached different stages:

8 processors: $t = 0.122$ s 64 processors: $t = 0.139$ s 512 processors: $t = 0.189$ s

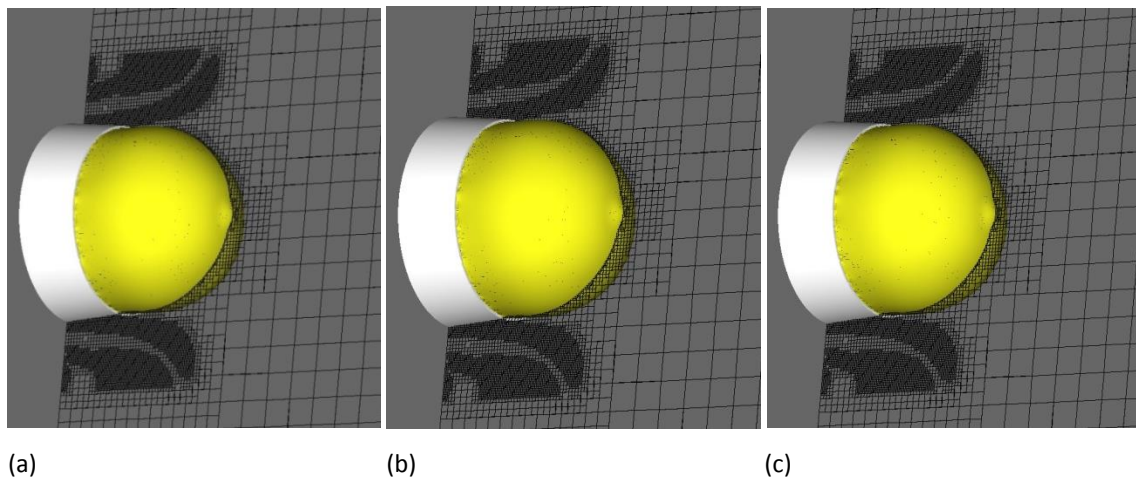


Figure 11 Snapshot of the mesh, the solid and the liquid interface at $t = 0.122$ s at level 9: (a) using 8 processors, (b) using 64 processors and (c) using 512 processors.

In order to make a decision about how many cores should be used, some graphics were made with the data obtained. Figure 12 shows the number of cells per second (Z) over the number of processors (np) as function of the timestep depending on the number of processors used. It can be easily seen that using 512 cores the speed is much slower than in the other two cases. Not so obvious is that the job with 64 processors tends to be slightly faster than the one executed with 8.

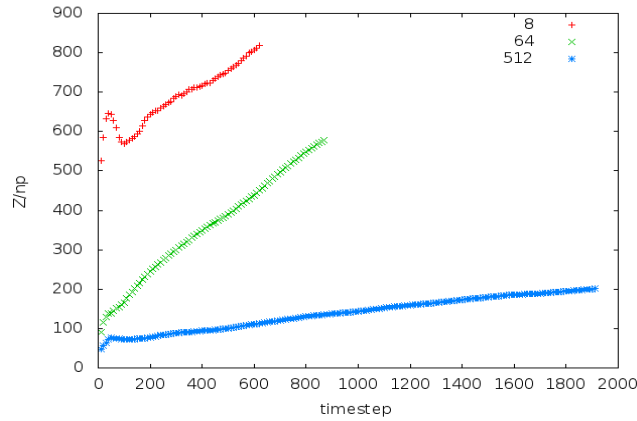


Figure 12 Z over np as a function of the timestep for case A at level 9

Another aspect of the simulations that deserves to be considered is the balance between the cores. Since in all cases the number of processors is lower than the number of boxes, each processor has a concrete number of boxes assigned at any time. However, this distribution is not always as equitable as desired, and this could be a reason to decide whether a simulation is efficient or not. The datum which tells that information is the difference between the maximum and the minimum boxes per processor: as smaller the difference is, better is the efficiency.

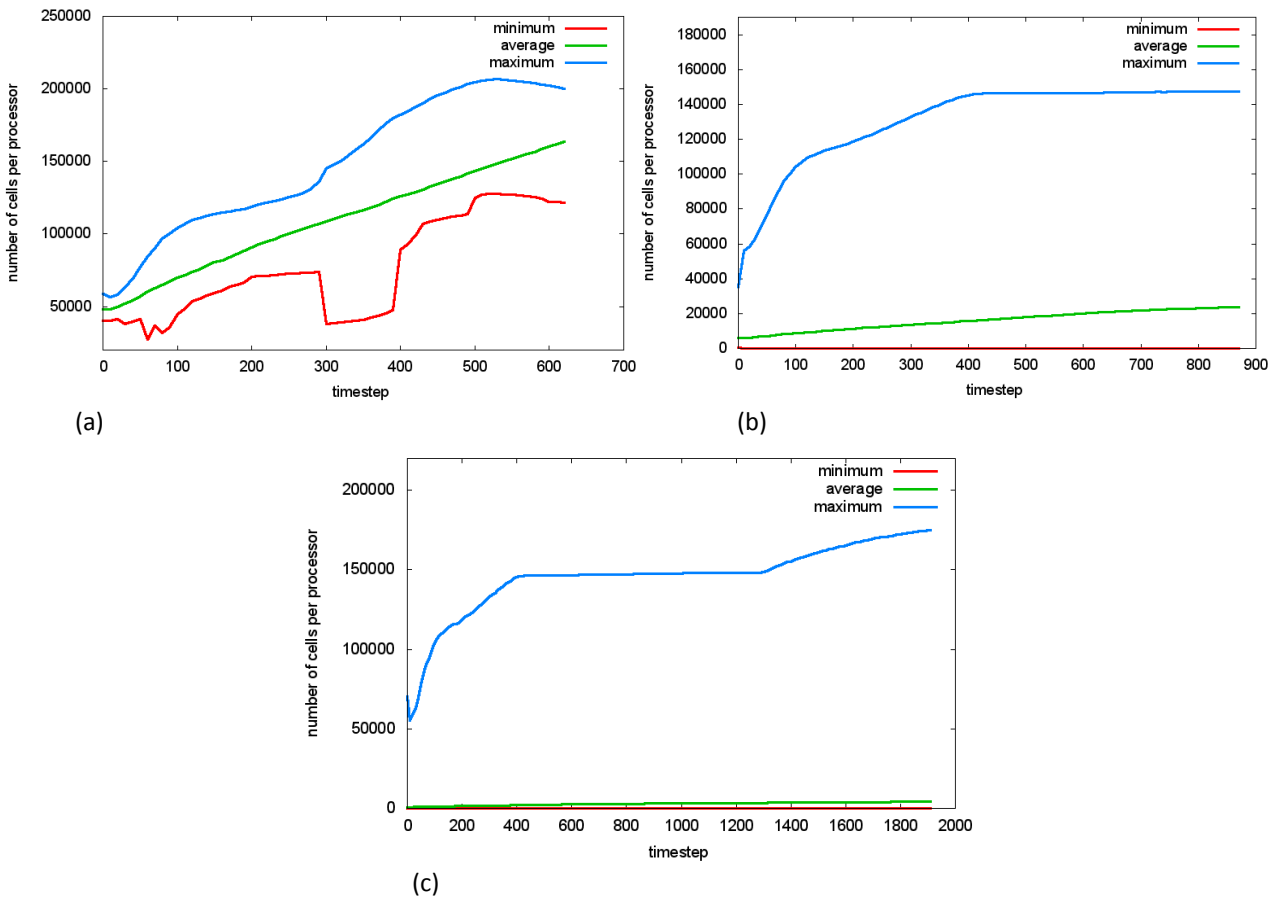


Figure 13 Number of cells per processor as a function of the timestep on 8 (a), 64 (b) and 512 (c) cores.

For the current case, the maximum, the average and the minimum number of processors as a function of the timestep are presented depending on the number of cores used to carry out the simulation. Looking at Figure 13 it can be easily remarked that the difference between the maximum and the minimum number of boxes per core is much smaller in the case where less processors are used to execute the simulation.

Even though it could seem that the best option would be to run the simulation on 8 processors, it was finally decided to do it on 64. Although the balance seems to be much better for the 8 cores case, it has to be considered that it can vary significantly from the very beginning stages to the end of the simulation. Since the Figure 14 was done after 24 hours simulating, it was assumed that as the simulation would progress the balance for the 64 cores case would get much better. Furthermore, as mentioned before, the Figure 13 shows that the 64 cores case is slightly faster than the case which uses 8 cores. The job executed with 512 was not considered in the decision since all the results were notably weaker compared to the other two.

Before proceeding to make more changes on the simulation in order to get smoother results, the current one was submitted to run for 5 days on 64 processors so as to check the results at further stages. After 5 days the last stage reached was at $t = 0.345$ s.

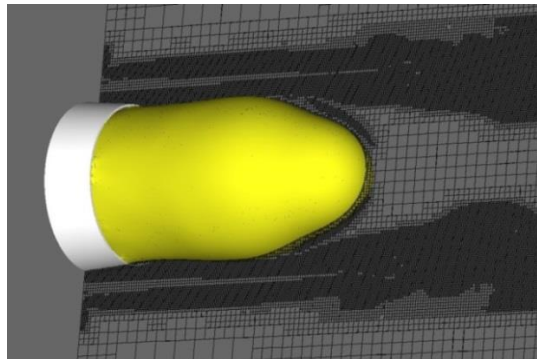


Figure 14 Snapshot of the mesh, the solid and the liquid interface at $t = 0.345$ s for 64 cores at level 9

6.1.2. Case A at level 10

After having checked that the simulation succeeded to run on Jade, it was time to improve it. In order to obtain further resolved results, it was decided to increase the level from 9 to 10. Although in the section above it was realized that on 64 processors the simulation was more efficient than in 8 or 512, there was a probability that it was not the case at level 10. Hence, the three cases were also executed at level 10 so as to compare the results and make a choice on the most competent one.

As it could be expected, increasing the level to acquire higher resolution was not at any price.

After 24 hours simulating each case reached the following stage:

8 processors: $t = 0.72 \text{ s}$ 64 processors: $t = 0.107 \text{ s}$ 512 processors: $t = 0.131 \text{ s}$

It is easy to see that the final stage after 24 hours simulating is an earlier step compared to the same simulations executed at level 9. Although the rise of the level leads to a further refined mesh and smoother result, it is not for free, and the speed of the simulations decreases.

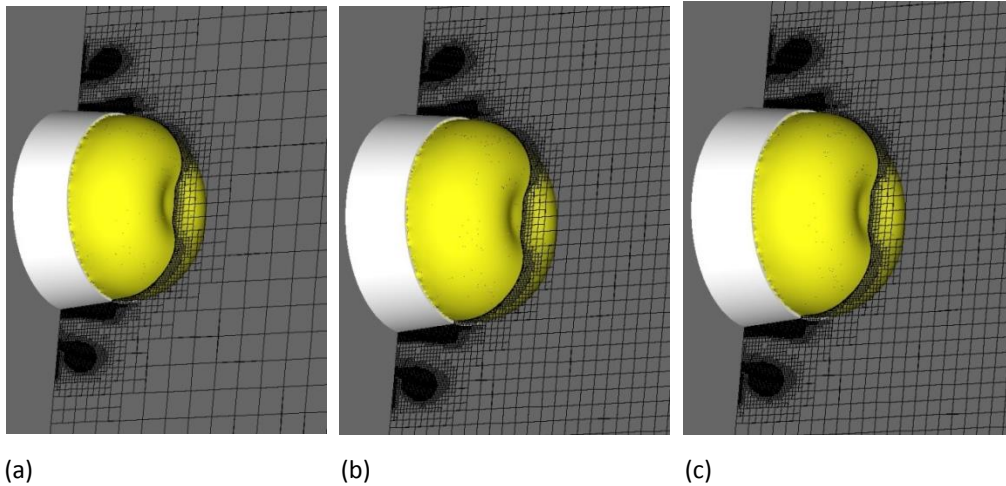


Figure 15 Snapshot of the mesh, the solid and the liquid interface at $t = 0.72 \text{ s}$ for level 10 using 8 (a), 64 (b) and 512 (c) processors.

On the other hand, the speed tendency depending on the number of cores used for the simulation does not differ much when increasing the level. Looking at Figure 16 it can be seen that executing the job on 512 cores leads to a very slow simulation (as in Figure 12).

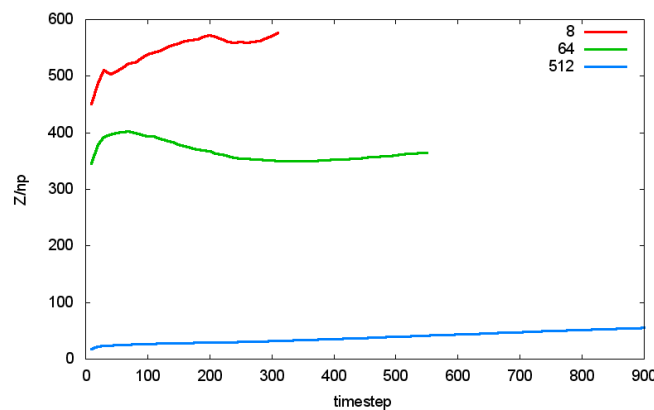


Figure 16 Z over np as a function of the timestep for case A at level 9

Concerning the balance between the number of cells that each processor takes care of, having a look at Figure 17 it can be observed that, the difference between the maximum and the minimum number of cells per processor is directly proportional to the number of processors.

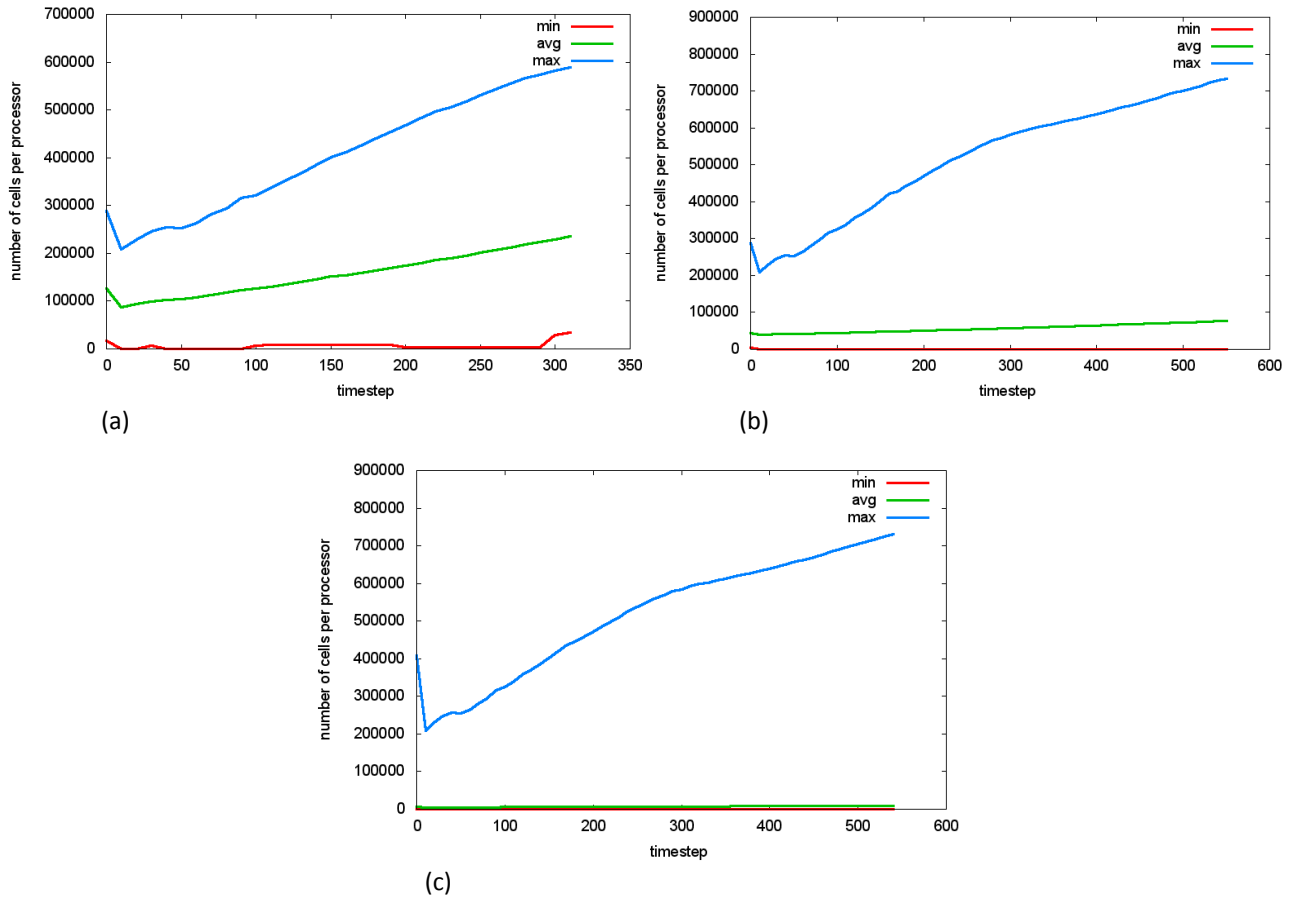
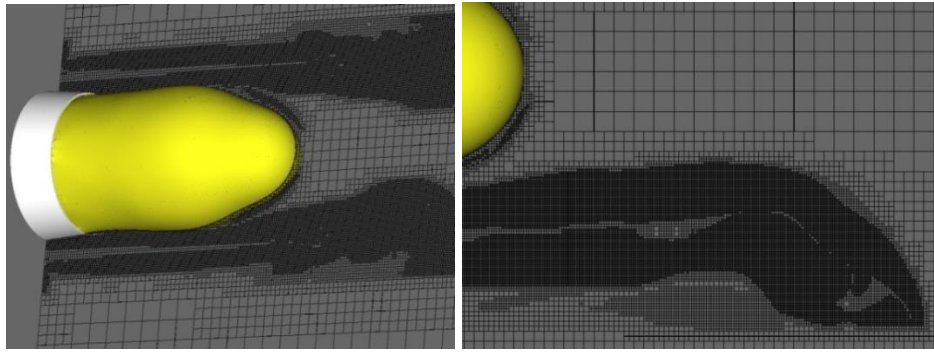


Figure 17 Number of cells per processor as a function of the timestep on 8 (a), 64 (b) and 512 (c) cores.

In the same way as the previous case (case A at level 9), it could seem that running the job on 8 processors may result in the most efficient simulating. Nevertheless, it has to be taken into account that Figure 17 (a), (b) and (c) were done after 24 hours simulating, and that the results might differ significantly from the beginning to the final states. Consequently, considering that in the final stages much more boxes would be occupied by the liquid, it was thought that running the simulation on 64 cores would be more efficient than doing it on 8.

Once decided to execute the simulation on 64 processors, the job was restarted from the last file acquired after those 24 hours and submitted to simulate for 120 hours¹⁶. This procedure was repeated until the job has been simulating for about 50-55 days with a last state at $t = 0.379$ s. At this stage the x-component of the velocity (U) reaches a maximum value of 23.19 m/s, while the minimum is of -3.2 m/s (dark blue area of Figure 19).

¹⁶ Maximum number of hours permitted by JADE.



(a) (b)
Figure 18 Snapshot of the mesh, the solid and the liquid interface at $t = 0.379$ s on 64 cores at level 10

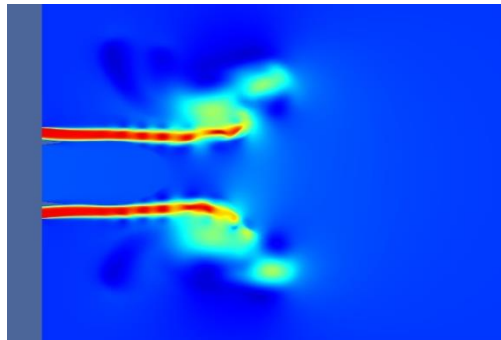
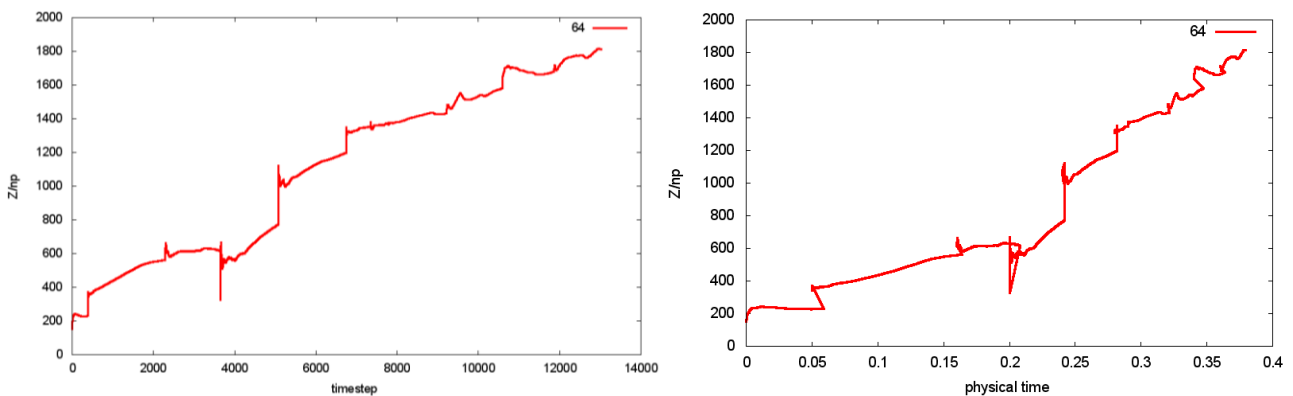


Figure 19 X-component of the velocity at $t = 0.379$ s on 64 processors at level 10

As might be expected, the speed of the simulation (number of cells per processor per second) increased at the time that the simulation advanced. In addition, it was noticed that usually when a restart was done, the speed suffered a notable peak (look at Figure 20) which led to a further augment of the velocity.

On the other hand, it can be also observed that at the end of the simulation the difference between the maximum and the minimum number of cells became smaller, which means that at the time that the simulation progressed, the balance between the different processors turned out to be better (look at Figure 21).



(a) (b)
Figure 20 Z over np for case A at level 10: (a) as function of the timestep, (b) as function of the physical

time

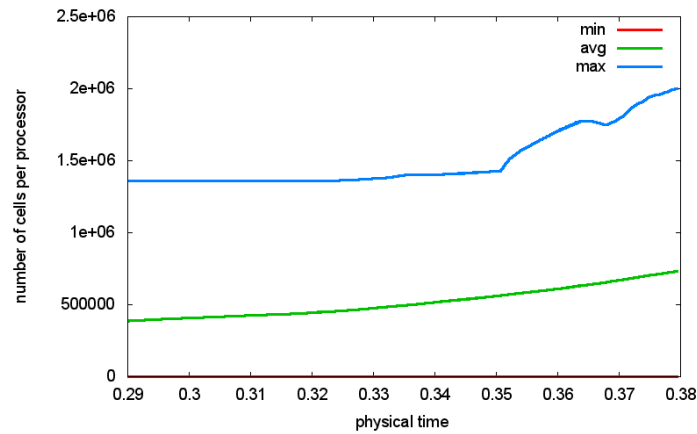


Figure 21 Number of cells per processor as a function of the physical time on 64 cores

Another variable that conditions the speed of the simulation is the dt , that is, the time between every step. Looking at Figure 22 it can be seen that the timestep decreases as time progresses, which mean that the simulation velocity is lower every time.

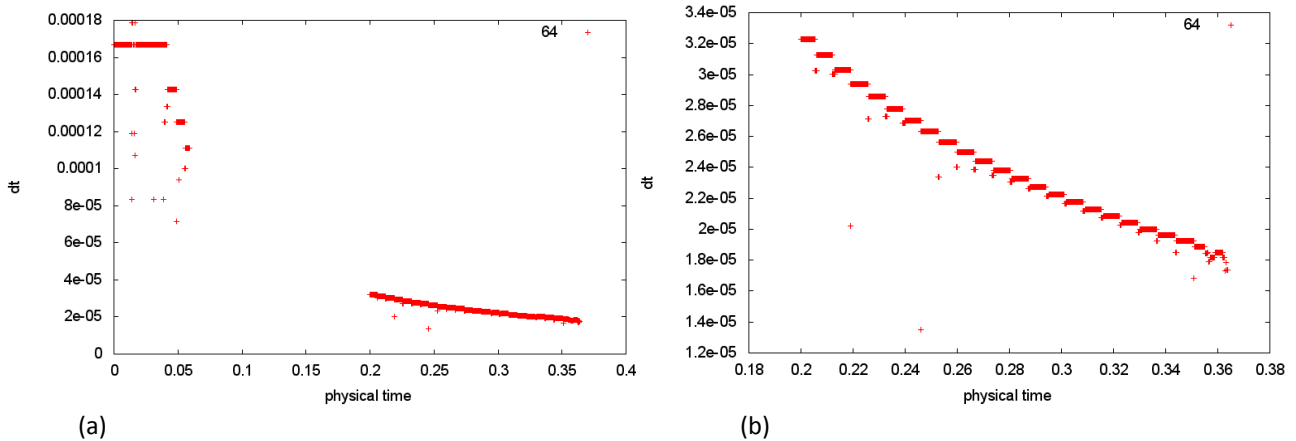


Figure 22 Dt as a function of the physical time

6.2. Case B

In this case, the simulation was meant to run for 5 days at level 9 but, it could not be said surprisingly, it was stopped by itself after around 72 hours. It was checked that at the last stages before the stop some droplets were starting to be created, which was blamed to be the reason of instabilities that caused the finish.

In order to attempt to resolve this obstacle, the simulation was restarted at a higher level (from 9 to 10) from the last stage where the liquid interface was still completely smooth ($t = 0.280$ s). However, the simulation was stopped by itself once again even before getting to the last state that had been reached at the lower level.

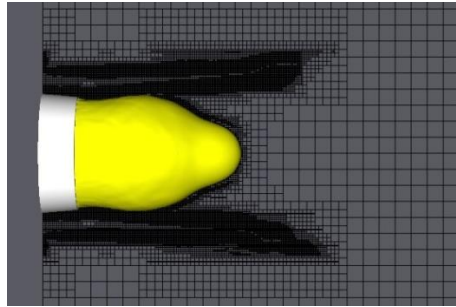
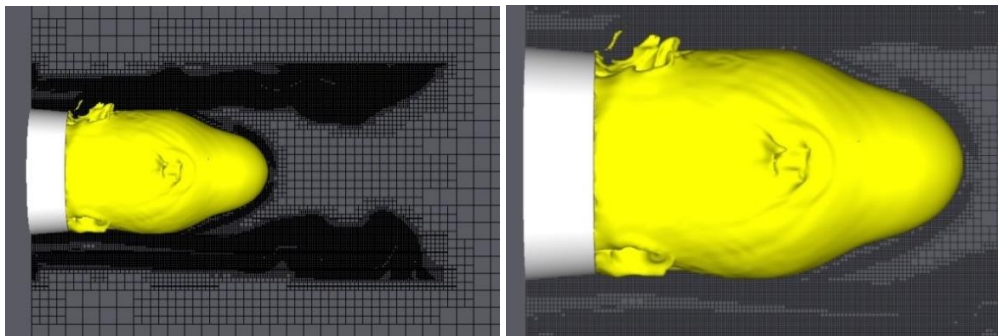


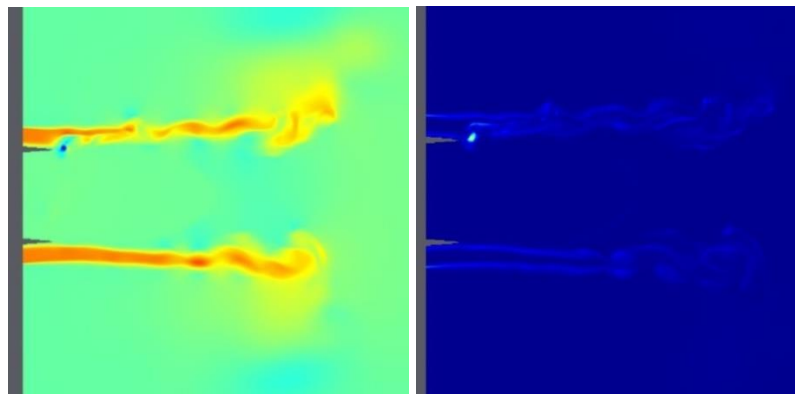
Figure 23 Snapshot of the mesh, the solid and the liquid interface at $t = 0.280$ s with level 9



(a)

(b)

Figure 24 Snapshot of the mesh, the solid and the liquid interface at $t = 0.341$ s with level 9



(a)

(b)

Figure 25 X-component of the velocity (a) and vorticity field (b) at $t = 0.341$ s with level 9

Looking at Figure 25 (a) it can be observed that the higher negative velocity (dark blue section), which has a value of -30.9 m/s, is exactly situated in the location where a bubble is detached (look at Figure 29). A similar phenomenon happens with the vorticity, since the greatest vortice with a vorticity of $1.031 \cdot 10^{-4} \text{ s}^{-1}$ is positioned in the identical region.

The appearance of those instabilities (bubbles, filaments and scams) was appointed to be the essential cause of the simulation stop itself. Due to the fact that before it had neither been able to run this simulation until the end, and knowing that even changing the whole parameter file maybe the problem would remain unresolved, it was decided not to spend more time on this case.

6.3. Case C

This case was first launched at level 7 in order to get fully resolved results quickly, and after 5 days ($t = 0.333$ s) simulating the liquid jet already showed filaments and bubbles getting away from the liquid jet as well as thread breakup (look at Figure 26).

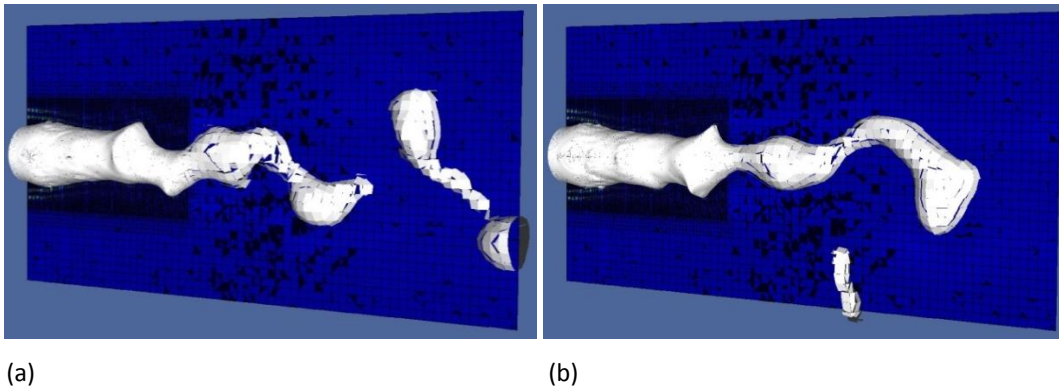


Figure 26 Snapshot of the mesh, the vorticity and the liquid interface at level 7: (a) at $t = 0.125$ s, (b) at $t = 0.333$ s

Nevertheless, it was remarked (look at Figure 26) that the mesh was only refined in a limited zone of the whole domain. Hence, the code was modified to widen the refined mesh region, at the same time that the level was increased to obtain smoother results, although it caused a descent of the simulation speed. The adjusted job was run for about 9 days and the simulation got to $t = 0.025$ s.

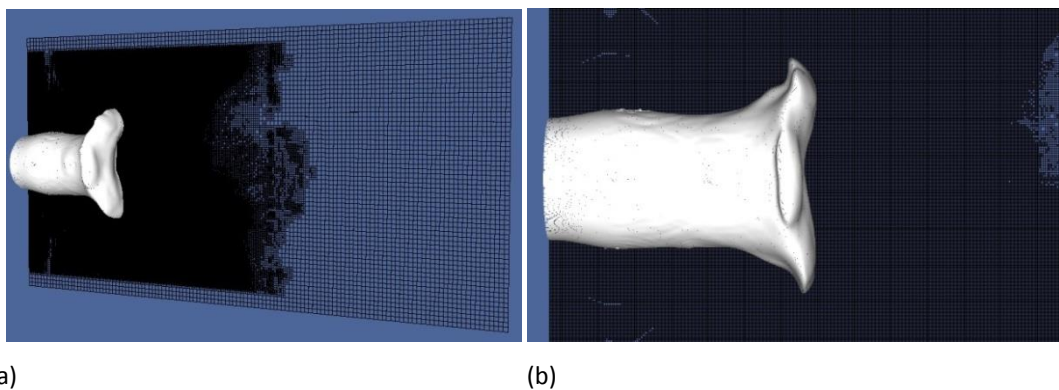


Figure 27 Snapshot of the mesh and the liquid interface at $t = 0.025$ s

This time no thread breakup or filament formation was observed. Nevertheless, some instabilities show that they would have appeared soon in the same way as before if more time had been had. This can also be remarked in the flow structure around the jet. Both the X-component of the velocity and the vorticity fields present circumstantial evidence of instability and vortices formation, which precede breakup (look at Figure 28). At the last stage of the simulation ($t = 0.025$ s), the X-component of the velocity (U) reached a maximum value of 526.3

m/s, which is actually the value imposed in the parameters.

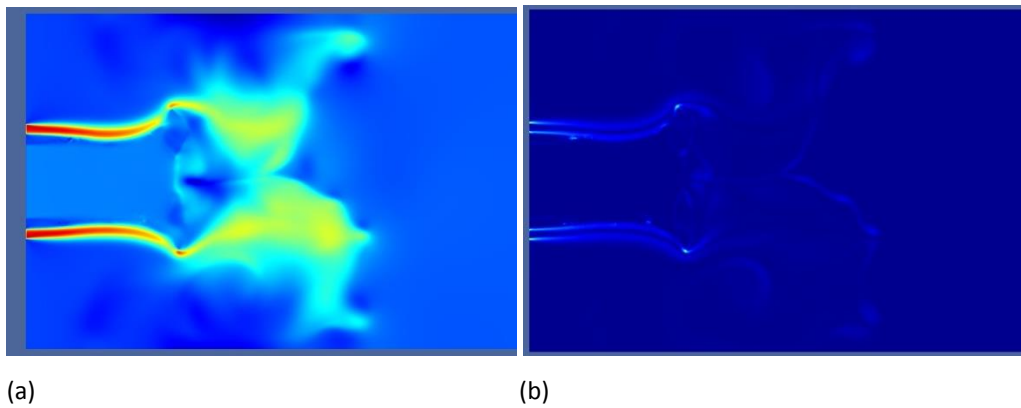


Figure 28 Snapshot at $t = 0.025$ s: (a) X-component of the velocity, (b) Vorticity.

Referring to the balance, the average of the number of cells per processor increases faster in this case rather than in case A and B. This is believed to be due to fact that the liquid advances faster and more boxes are occupied, so the distribution of the cells becomes more equitable.

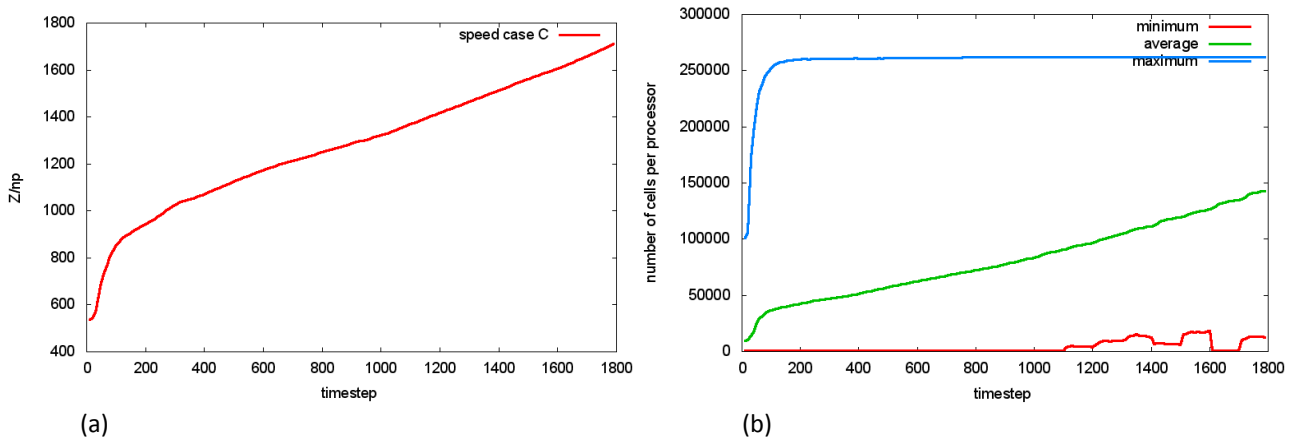


Figure 29 (a) Z over np as a function of the timestep; (b) Number of cells per processor as a function of the timestep.

6.4. Comparison

With the results presented in the previous sections comparisons can be made between case A and case C.

	Case A	Case C
M	3.516	7.346
We_0	157.978	28.150
Re_{gas,δ_g}	339.606	160

Table 9 Comparative table of the most important dimensionless numbers of cases A and C.

It is easy to see that case C converges to atomize much faster than case A. To get to a similar

liquid jet length in relation to the nozzle dimension, while for case A it takes $t = 0.260$ s, for case C it only takes $t = 0.012$ s. This means that case C is almost twenty times faster than case A.

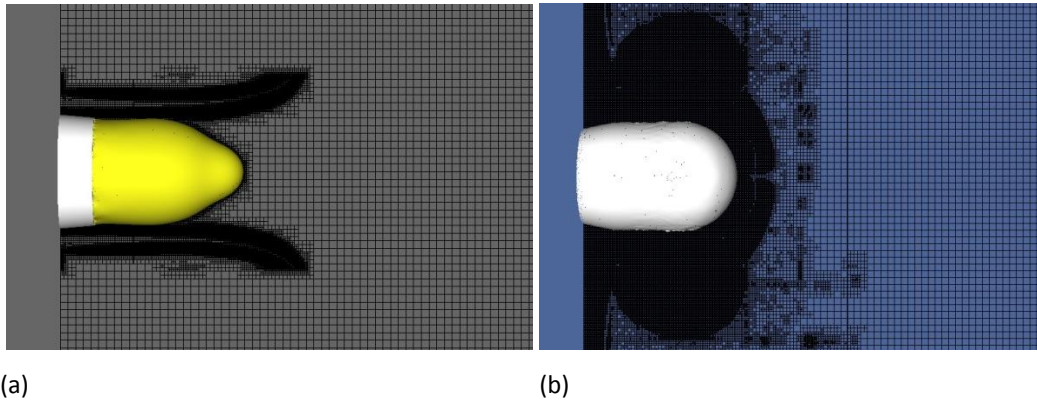


Figure 30 Snapshot of the mesh, the solid and the liquid interface: (a) case A at $t=0.260$ s, (b) case C at $t = 0.012$ s.

The fact that case C converges to atomize before than case A is due to the fact that, even though M is lower for case A than for case C, both the We_0 and the Re_{gas,δ_g} are much higher. Greater Re_{gas,δ_g} results from larger δ_g , and that means less shear viscosity.

On the other hand, it also results of the fact that the gas velocity is governed by the error function $erf(z)$ ¹⁷, which depends on different values for the different cases.

$$\begin{aligned}
 & \text{Case A} \\
 & U_g(t) = erf(t/1) \\
 & \text{Eq. 23}
 \end{aligned}$$

$$\begin{aligned}
 & \text{Case C} \\
 & U_g(t) = erf(t/0.01) \\
 & \text{Eq. 24}
 \end{aligned}$$

Looking at Eq. 23 and 24 it is easy to see that the gas will achieve its maximum velocity much later in case A than in case C.

¹⁷ $erf(z)$ is the "error function" encountered in integrating the normal distribution (which is a normalized form of the Gaussian function). It is an entire function defined by

$$erf(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.$$

6.5. Gerris performance

6.5.1. Real Solid vs. Virtual Solid

As it has been specified before, while case A and B were run using a real solid, case C was defined with a virtual solid. The reason of this modification was not only because some problems came up when trying to launch it with a real solid, but also because it had been observed that using a virtual solid instead of a real one turned out into a faster simulation speed (look at Figure 31).

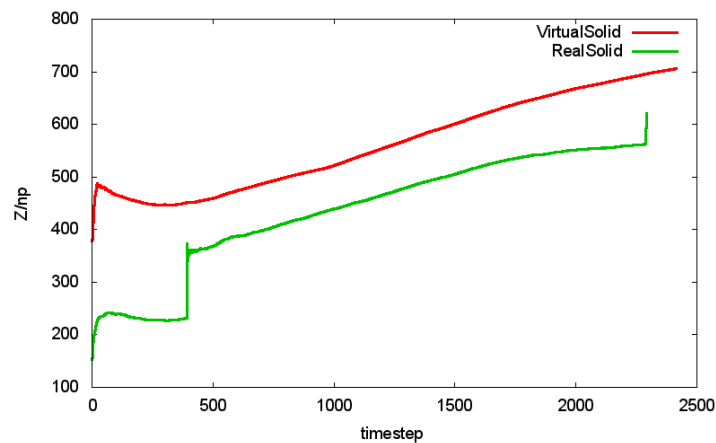


Figure 31 Z over np as a function of the timestep depending on the type of solid, for case A on 64 cores.

6.5.2. Jade 1 vs. Jade 2

Actually, the supercomputer Jade has two units. The Unit 1 of the Jade machine (Jade 1) consists of 1536 nodes. On each node, two processors of the Harpertown type (Intel Quad-Core E5472) are available with 4GB of memory per core. On the other hand, the Unit 2 of the Jade machine (Jade 2) is composed of 1344 nodes. On each node, two Nehalem type processors (Intel Quad-Core X5560) are available with 4GB of memory per core. [1]

All the simulations were launched on Jade 1 since Gerris had been only installed in this unit. Nevertheless, on the last days of the internship the CINES engineers succeeded to install it on Jade 2. In order to test it and compare it with Jade 1, a job that had been submitted on Jade 1 was submitted on Jade 2, and it turned out that the speed of the simulation was significantly higher (look at Figure 32).

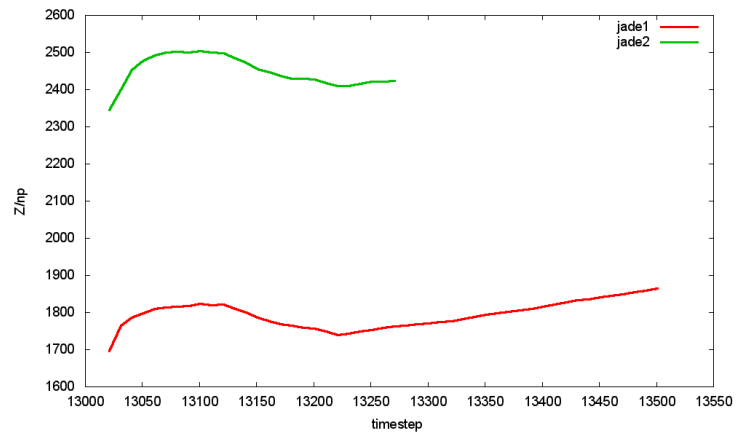


Figure 32 Z over np as a function of the timestep depending on the cluster, for case A on 64 cores.

CONCLUSIONS AND NEXT STEPS

Although none of the cases have atomized at the end of the internship, mostly due to the lack of time, sundry other interesting results have been obtained.

The most relevant conclusions have to do with the way with which the simulations speed increases. On the one hand, it has been observed that when describing the solid in the code, it is preferable to use a virtual solid rather than a real one. On the other hand, it has been proved that running the jobs on Jade 2 results in a higher simulation velocity compared to launching them on Jade 1.

In reference to the number of processors used to run the simulations, it was noticed that using more cores does not always mean to increase the efficiency of the simulation. Actually, it was discerned that the speed of the simulations was much lower when employing 512 processors than when using 8 or 64 cores.

As it has been said, the simulations didn't get to the desired state basically as a result of the lack of time, so the crucial next steps to follow the current project would be to continue the simulations for a significant period of time until they atomize.

BIBLIOGRAPHY

- [1] CINES. Le supercalculateur JADE. [<https://www.cines.fr/calcul/materiels-et-logiciels/le-supercalculateur-jade/>, February – July 2014]
- [2] Coleman, G. N. and Sandberg, R. D. *A primer on Direct Numerical Simulation of Turbulence – Methods, Procedures and Guidelines*. ePrints Soton – University of Southampton (School of Engineering Sciences). [http://eprints.soton.ac.uk/66182/1/A_primer_on_DNS.pdf, 16 June 2014]
- [3] Fuster, D., Matas, J. P., Marty, S., Popinet, S., Hoepffner, J., Cartellier, A., Zaleski, S. *Instability regimes in the primary breakup region of planar coflowing sheets*. *Journal of Fluid Mechanics* (2013), Vol. 736, pp. 150-176.
- [4] Gerris Flow Solver. [http://gfs.sourceforge.net/wiki/index.php/Main_Page, February – July 2014]
- [5] Institut Jean le Rond d'Alembert. [<http://www.dalembert.upmc.fr/ijlrda/>, February – July 2014]
- [6] Laseheras, J. C., Villermaux, E., and Hopfinger, E.J. *Break-up and atomization of a round water jet by a high-speed annular air jet*. *Journal of Fluid Mechanics* (1998), Vol. 357, pp. 351-379.
- [7] Marmottant, M., and Villermaux, E. *On spray formation*. *Journal of Fluid Mechanics* (2004), Vol. 498, pp. 73-111.
- [8] Osher, S., and Fedkiw, R. P. *Level set methods: an overview and some results*. *Journal of Computational Physics* (2001), Vol. 160, pp. 463-502.
- [9] Popinet, S. *An accurate adaptative solver for surface-tension-driven interfacial flows*. *Journal of Computational Physics* (2009), Vol. 228, pp. 5838-5866.
- [10] Popinet, S. *Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries*. *Journal of Computational Physics* (2003), Vol. 109 (2), pp. 572-600.
- [11] Tryggvason, G., Scardovelli, R., and Zaleski, S. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge: Cambridge University Press, 2011.
- [12] Versteeg, H. K., and Malalsekera, W. *An Introduction to Computational Fluid Dynamics (Second Edition)*. Glasgow: Pearson Education Limited, 2007.
- [13] WILCOX, D.C. *Turbulence Modelling for CFD*. California: DCW Industries, 1994.