Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university, or other institution of learning.

Acknowledgements (1)

I would like to thank here the people who were scientifically and/or administratively involved in this PhD work:

- En premier lieu, M. Damien VIOLEAU, ingénieur-chercheur au LNHE : je salue ici l'excellent pédagogue qui a assuré un encadrement de qualité au cours de ces trois années. Sa disponibilité et sa grande rigueur m'ont été largement bénéfiques.

- Je tiens à remercier chaleureusement Dr. Charles MOULINEC de l'Université de Manchester : son expertise et sa longue expérience en CFD ainsi que ses nombreuses compétences informatiques m'ont été d'une aide remarquable.

- Ik wil ook graag Dr. Bendiks Jan BOERSMA bedanken. Hij heeft mij heel veel geholpen tijdens het laaste jaar van mij promotie op de TU-Delft. Ik heb heel veel van Bendiks Jan geleerd wat de numerieke en theoretische aspekten van CFD betreft.

- I thank Professor Dominique LAURENCE (UMIST / EDF) for his technical help: his contribution to the published papers and the pre reading of my thesis were very useful. Moreover, I also appreciated his Marie-Curie scholarship management.

- Ik wil ook graag Dr. Rene DELFOS (TU-Delft) bedanken. Hij heeft mij heel veel met het experimentele werk van de dam breaking geholpen.
  Tenslotte vergeet ik niet Professor Frans NIEUWSTADT (TU-Delft), die mij veel gemene vragen heeft gestelt tijdens de lunchpraatjes.

- Last but not least, I thank three secretaries who did a very efficient administrative work during these three years: Miss Patricia SHEPHERD from UMIST, Miss Ria VAN DER BRUGGE-PEETERS from TU-Delft and Miss Christine POUVIOT from EDF.

Acknowledgements (2)

During the last three years, I had the opportunity to work in three countries, which implies different cultures and habits. I hence met several people with who I learnt a lot and due to them, these three years were full of experiments.

- Je salue ici les acteurs du groupe P75 du LNHE d'EDF et particulièrement Thierry BOULET et sa bonne humeur contagieuse (surtout sur un court de tennis), Arnaud LENES qui avait pour habitude d'occuper le bureau K125 dix minutes avant la pause de 16h, Emile RAZAFINDRAKOTO ou le professeur Tournesol de CFD et Damien VIOLEAU, pilier du Vieux-Paris.
- Now, I would like to thank some people who helped me to support the sunny Manchester: Mike DEEVY and Simon GANT, with who it was a pleasure to discuss, Juan URIBE (Termita sote) for the friday and squash sessions, Eun-Sug LEE (My Dear Nuna) who used to upset me with her chocolate and sleeping obsessions. A special thought for Vassillis KARANIKOLIS, for all the discussions we had in B4 and in other places, like in front of Marks and Spencer. Finally, Jason BASSI, Guilhem and Aurélie LAURANS (Frog Couple) with who I had some great laugh sessions.
- Finally, I would like to thank all the labmembers of Aero and Hydrodynamica of TU-Delft, for the outings, the drink sessions, the BBQs, ...

Acknowledgements(3)

Je dédie cette thèse

- A mes parents, qui m'ont toujours tout donné.
- A Aine, mon compagnon de vie.
- A ma princesse Crune, avec qui je partage une complicité empreinte d'une harmonie indicible. Tes joies et tes rires, ainsi que tes tourments et tes angoisses sont les mêmes que les miens à quinze ans d'intervalle, Vié Bébète.

Je salue ici mes relations privilégiées

- Marion JEANNE, mon amie de coeur.
- Céline OLIVIER, ma maîtresse préférée.
- Ariane DAVID, qui me comprend si bien.

Aux véritables amitiés qui se sont forgées au cours de ces trois années : Constantinos ZEGOS, Victor ROSALES SIERRA, Toshiaki FUJII et Charles MOULINEC.

Abstract

Smoothed Particle Hydrodynamics (SPH) is a fully gridless Lagrangian method, invented in 1977 for astrophysical simulations. Due to its simplicity and robutness, this numerical method has been extended to solid mechanics for rapid dynamic phenomena and more recently to complex fluid mechanics problems such as dam breaking and wave flumes. However, the literature is quite scarce regarding SPH "academic" flow validation and turbulent flows modelled with SPH.

The first goal of this work is to validate SPH for "classical" laminar problems. A complete investigation of a 2D laminar free surface channel flow is presented: the influence of the viscous term modelling and SPH parameters as the smoothing length are discussed. Moreover, a new practical wall modelling is introduced, as well as no-slip boundary conditions. Velocity and pressure profiles are in excellent agreement with the theory. SPH simulations of 2D laminar separated flows are also considered through a periodic hill channel and a backward facing step geometry. Separation and reattachment points are fairly well predicted and velocity profiles are consistent with those relative to Eulerian codes. A limitation of the nearly incompressible assumption used in most standard SPH codes is also revealed.

The second goal of this thesis is to extend SPH to turbulence modelling. A SPH mixing length model is introduced and applied to a 2D turbulent free surface channel. The velocity profiles are in good agreement with the theoretical ones, for smooth and rough beds. However, an overestimation of the axial velocity gradient in the vicinity of the wall causes a slight overestimation of the eddy viscosity profiles.

In order to adapt Large Eddy Simulation (LES) concept to SPH, a 3D code was at first developed. The implementation of a simple Smagorinsky model is described and some first results relative to a 3D turbulent free surface channel are presented. Typical turbulent fields with velocity fluctuations in each direction are obtained. Moreover, averaged pressure and velocity profiles are on the whole fairly consistent with the theoretical ones. The results are promising but more numerical work has to be achieved to reduce the turbulence intensity and the computing times. A simulation of a 3D dam breaking is also presented: SPH results and experiments are qualitatively in good agreement.

Contents

Chapter 1

Introduction to Particle and Meshless Numerical Methods

The analytical study of fluid mechanics acknowledges two descriptions of fluid flow, Eulerian and Lagrangian:

- In the Eulerian description, one concentrates on what happens at a fixed spatial point $\underline{x}$ at a time t. Here, the independent variables are taken as $\underline{x}$ and t. Any flow variable F is then expressed as $F\left(\underline{x}, t\right)$.

- In the Lagrangian description, one follows the history of an individual fluid parcel through the time. In this description, a flow variable is characterised by two independent variables which are the particle label and the time t. Usually, as the particle label, the point vector $\underline{x}_0$ of the particle at the reference time $t = 0$ is chosen. In the Lagrangian description, any flow variable F is thus expressed as $F\left(\underline{x}_0, t\right)$.

The implementation of Computational Fluid Dynamics (CFD) in engineering applications is most of the time based on the Eulerian description of the flow, since these techniques have been heavily studied for fifty years and are clearly understood. Most of the commercial codes have been developed using finite volume (Fluent, Star-CD, Comet, Saturne...), finite element (Pronto, Telemac, ...) or finite difference approaches. The result is that, unfortunately, Eulerian based models are used even for applications for which they are not suited: for instance, large deformations and complex free surface flows. That is the reason why Fox and McDonald, in their introductory fluid mechanics text book [22], state “Clearly the type of analysis depends on the problem. Where it is easy to keep track of identifiable elements of mass, we use a method of description that follows the particle”. Implicitly, this means “particle methods”.

1.1 Particle methods

Over the past 10 years, the trend in CFD has been to focus on increasingly demanding problems that require the ability to treat large deformations, complex geometry, non-Newtonian fluid behavior, discontinuities and singularities. This trend finds applications in the simulation of industrial and manufacturing processes such as extrusion where extremely large deformations occur, or in the simulation of flows within a porous medium. Although adaptive techniques and remeshing strategies have been developed for these classes of problems, the most viable strategies are so-called particle methods. Since their invention by Harlow [29] more than 50 years ago, particle methods have shown their huge potential in many applications: plasma simulation depends almost exclusively on the Particle-In-Cell method, the liquid drop model for nuclear fission can be considered by solving relativistic equations with particle methods and advection in atmospheric computations is modeled using semi-Lagrangian methods. Sea ice dynamics have also simulated with these methods [7]. Three widespread particle methods are briefly described in this part.

1.1.1 Marker-And-Cell method

The earliest particle numerical method devised for time-dependent, free-surface, flow problems was the Marker-And-Cell (MAC) method [30] . This technique is based on a fixed Eulerian grid of control volumes. The fluid location within the grid is determined by a set of marker particles that move according to locally interpolated fluid velocities, but otherwise have no volume, mass or other properties [73]. These marker particles can be interprated as centroids of small fluid elements. Grid cells containing markers are considered occupied by fluid, while those without markers are empty (or void). A free surface, for instance, is defined to exist in any grid cell that contains particles and that also has at least one neighboring grid cell that is void. An interesting feature of the MAC method is that the markers do not track surfaces directly, but track fluid volumes. Surfaces are simply the boundaries of the volumes, and in this sense surfaces may appear, merge or disappear as volumes break apart or coalesce. Some special treatments are then required to define the fluid properties in newly filled grid cells and to cancel values in cells that are emptied.

The extraordinary success of the MAC method in solving a wide range of complicated free-surface flow problems is well documented in numerous publications. Filling process into a cubic container for instance, as represented in figure 1.1[1], has been well modelled with the MAC method by Ji. Liovic [50] investigated splash resulting from top-submerged gas injection: the MAC model shows phenomena of bubble formation, bubble rise and splash drop formation in detail, as shown in figure 1.2. In spite of its success, the MAC method has primarily been used for two-dimensional simulations because it requires considerable memory and CPU time

[1] $http://www.me.uwaterloo.ca/\sim fslien/free\_surface/free\_surface.htm$

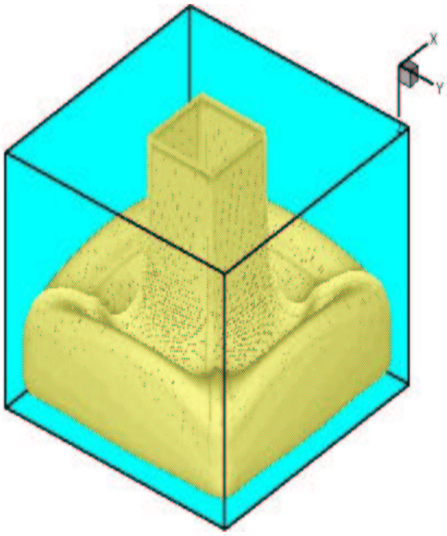

Figure 1.1: Filling process in a cubic container modelled with the MAC method by Ji.

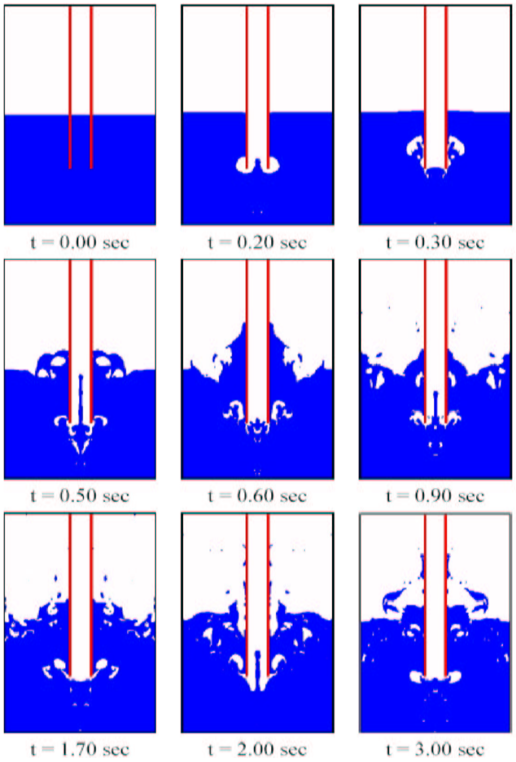

Figure 1.2: Simulation of a top-submerged gas injection in water with the MAC method [50].

to accommodate the necessary number of marker particles. Typically, an average of about 16 markers in each grid cell is needed to ensure an accurate tracking of fluid volumes undergoing large deformations. Moreover, the MAC method is more complicated to code than other particle methods.

1.1.2 Particle-In-Cell method

The Particle-In-Cell method (PIC) was developed by Harlow [28], [9], [8] in 1957 and soon applied in CFD. Like the MAC approach, PIC is a dual description method, Lagrangian and Eulerian. The main idea is to track the motion of a set of particles that carry velocity, position, mass and species information, in a Lagrangian manner. However, the momentum equation is solved on an Eulerian mesh, which provides a convenient pattern to define discrete derivatives: the particle variables are interpolated to this mesh and classical finite differences can then be used to discretise the momentum equation. The new values are re-interpolated to particles which move according to them.

PIC is applied in order to treat transient, compressible flows of multiple materials with no restriction on interfacial deformation. It is also the reference method for plasma simulations [7]. An accurate simulation of the Kelvin-Helmholtz instability has been performed by Lapenta (see figure 1.3) [42] with the PIC method. However, it is well known that the main drawback

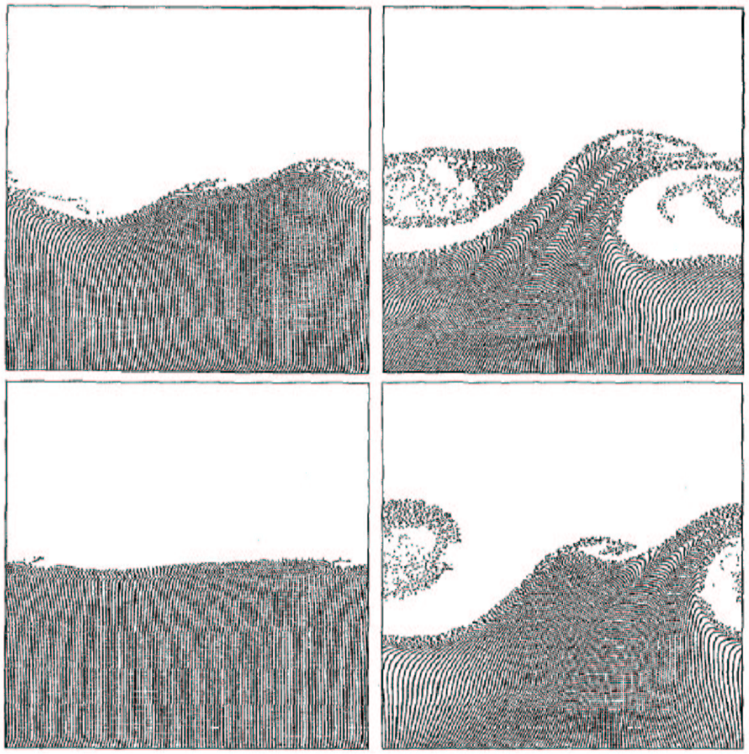

Figure 1.3: Kelvin-Helmholtz instability simulated by PIC: plot of the particle locations [42].

of such methods is that they are time consuming. Indeed, interpolation from the particles to the mesh and vice-versa is quite heavy in terms of CPU consideration. Ensuring the numerical stability of the interpolations and extrapolations is also quite delicate [21]. Furthermore, one must know the location of the regions where steep gradients occur to make the grid fine enough to resolve these structures.

1.1.3 The Lattice-Boltzmann Method

The Lattice-Boltzmann Method (LBM) uniformly discretises the space with hexagonal or square lattice in 2D and with cubic lattice in 3D. On such a discrete space, a macroscopic fluid is replaced with the population of mesoscopic fluid particles with unit mass, which are allowed to be rest at lattice nodes or to move with constant velocity along lattice lines, as shown in figure 1.4. LBM thus describes a physical system in terms of the motion of fictitious mesoscopic particles on a lattice. During one time step, two kinds of motion are repeated all over the space: translation from site to site, and elastic collision between particles at each lattice site. The collision is statistically operated according to rules ensuring mass and momentum conservation. As as result, a macroscopic fluid dynamics in LBM appears from averaging particle motions [78]. The Lattice-Boltzmann method has been particularly successful in simulations of flows involving complicated boundaries, viscoelastic fluids, particulate suspensions in fluid. In the same way, chemical reactive species, combustions, magnetohydrodynamics, crystallization, and other complex systems can be accurately modelled with LBM. It is also useful for computing fluid flow in complex geometries like random porous media. Figure 1.5 represents a flow through a porous medium, computed by Lin at the University of Iowa [2]. The Rayleigh-Taylor instability has also been well simulated by the Fuji Research Institute Corporation (see figure 1.6) with LBM [3]. Moreover, this technique is also efficient in terms of CPU and is easy to parallelize.

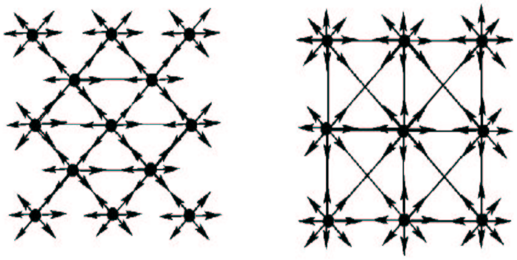

Figure 1.4: Velocity directions on a triangular lattice (left) and a square lattice (right) [49].

[2] *http : //www.icaen.uiowa.edu/ ∼ ching/resh_lbm.html*

[3] *http : //www.fuji − ric.co.jp/english/solution/prom/complex/kohno/lbmindex.html*

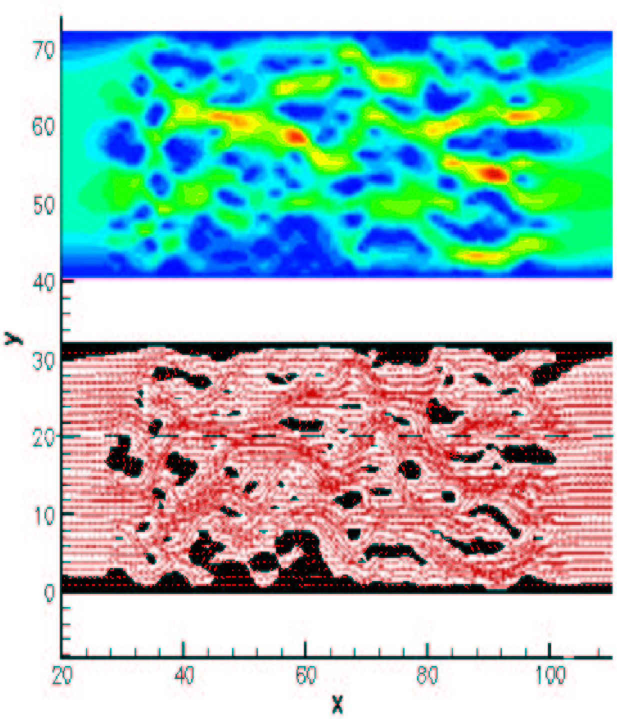

Figure 1.5: Flow through a porous medium simulated with LBM by Lin (University of Iowa): axial velocity contour (top picture) and velocity field (bottom picture).

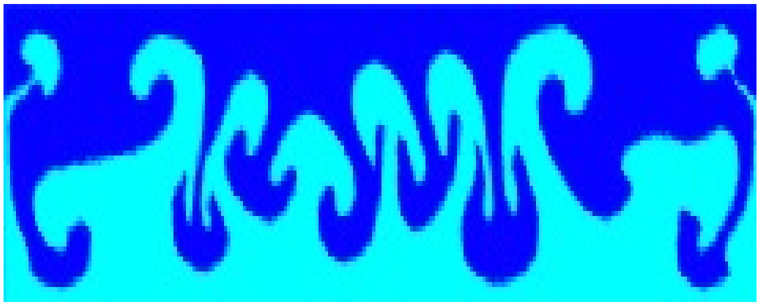

Figure 1.6: Rayleigh-Taylor instability modelled with the LBM by the Fuji Research Institute Corporation .

1.2 Meshfree methods

Meshfree methods eliminate the mesh relative to some particle methods and rely on a fully Lagrangian view of the problem. The relatively recent rebirth of interest in meshfree methods has led to a plethora of methods with a variety of names: DEM (diffuse element method), EFG (element free Galerkin), RKPM (reproducing kernel particle method), POU (partition of unity), PUFEM (partition of unity finite element method), MPM (material-point method), HP clouds, MLPG (meshfree local Petrov Galerkin), MLSPH (moving least squares SPH) and a host of other methods have emerged in the arena of meshfree methods. Two of them are briefly presented in this part: vortex methods and the Smoothed Particle Hydrodynamics (SPH) method.

1.2.1 Vortex Methods

In vortex methods, the Navier-Stokes equations are expressed in vorticity formulation and the vorticity field becomes the principal variable for computations [4]. The fluid velocity field is obtained from an integral of the vorticity, and the pressure is not explicitly solved for, as it is eliminated by the curl operator. Since vortex methods are characterised by a Lagrangian approach, fluid particles convect with the local fluid velocity at each time step.

Vortex methods are very efficient for investigation of problems characterised by high vorticity zones. For instance, Cottet [14] accurately revealed the density contours in a 2D shock-tube experiment with vortex methods: in figure 1.7, the initial density has a strong 100:1 discontinuity generating a shock wave that propagates to the right, then bounces back on the right wall and interacts with vorticity created in the boundary layer on the bottom wall.

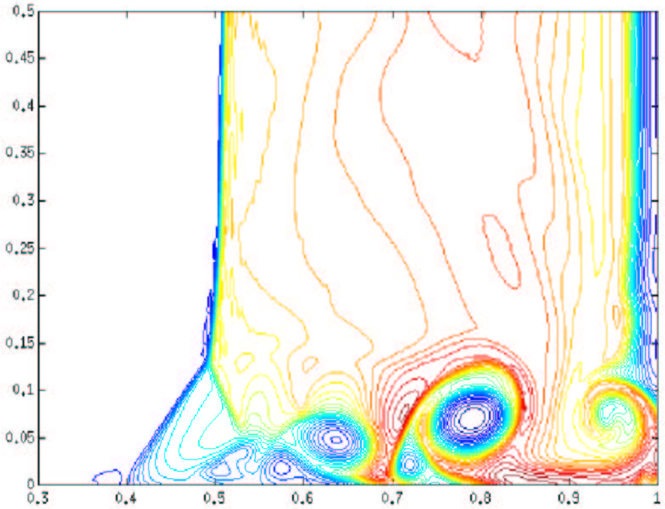

Figure 1.7: Density contours in a 2D shock-tube experiment realised with vortex methods [14].

Von Karman vortex streets behind a circular cylinder naturally develop 3D instabilities. When the cylinder is subject to rotation, the flow returns to a 2D topology. Vortex methods accurately simulate this phenomenon as well, as shown in figure 1.8 [15]. Vortex methods are not formulated to depend on a computational grid, and here lies one of their potential advantages, since grid generation is a very expensive process of CFD. In addition, the Lagrangian formulation is devoid of numerically diffusive truncation errors associated with the meshes [3]. However, the calculation of unsteady flows using the vortex method suffers from loss of accuracy due to the Lagrangian deformation of the particle field. As particles are allowed to freely convect, "gaps" appear in areas where the flow strain is large and thus the particle set loses the ability to reconstruct the smooth vorticity field accurately. Moreover, this method is not suited for compressible flows.

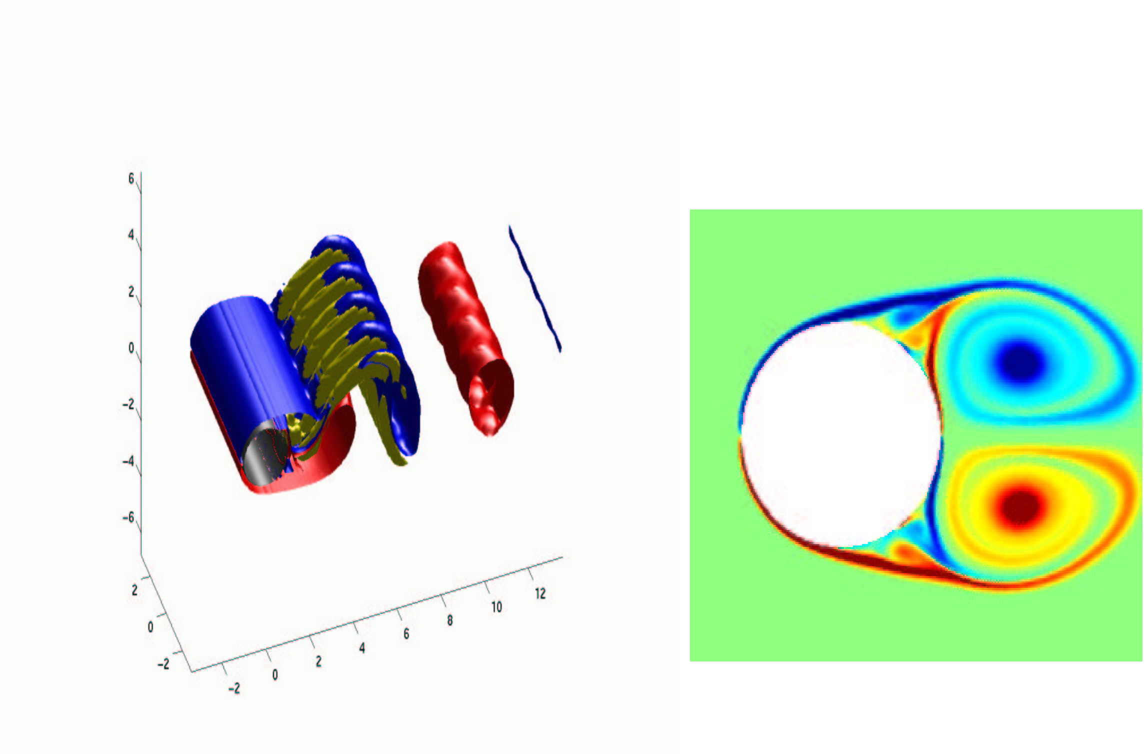

Figure 1.8: Von Karman streets simulated with vortex methods [15].

1.2.2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a meshfree Lagrangian method in which the medium is discretised by macroscopic particles. These individual particles interact with each other, move according to the medium motion and carry with them all of the physical information describing the medium. The technique is relatively new, having first been introduced by Lucy and Monaghan in 1977 [56], in the context of astrophysical modelling.

SPH in astrophysics

In a realistic and complete description of most cosmic phenomena, one must typically face highly non-linear interactions between objects or particles, as well as non-linear couplings between different kinds of interactions, including gravitational, electromagnetic, radiative, and gas dynamic interactions. Consequently, numerical approaches to understand astrophysical phenomena have become indispensable. Due to its meshfree feature, SPH is applied in astrophysics to simulate complex phenomena such as fragmentation within rotated clouds or supernovae explosions [56]. Indeed, classical Eulerian codes would require a very complex adaptative grid for this type of simulations. SPH has also been used to calculate star collision and galaxy formation [68]. Figure 1.9 shows some classical SPH applications in astrophysics, realised at Los Alamos National Laboratory [4]. Simulations of magneto hydrodynamic phenomena such as Alfven wave propagation, cosmological simulations of magnetic fields in galaxy clusters [16] and general relativity problems [56] are also successfully carried out with SPH.

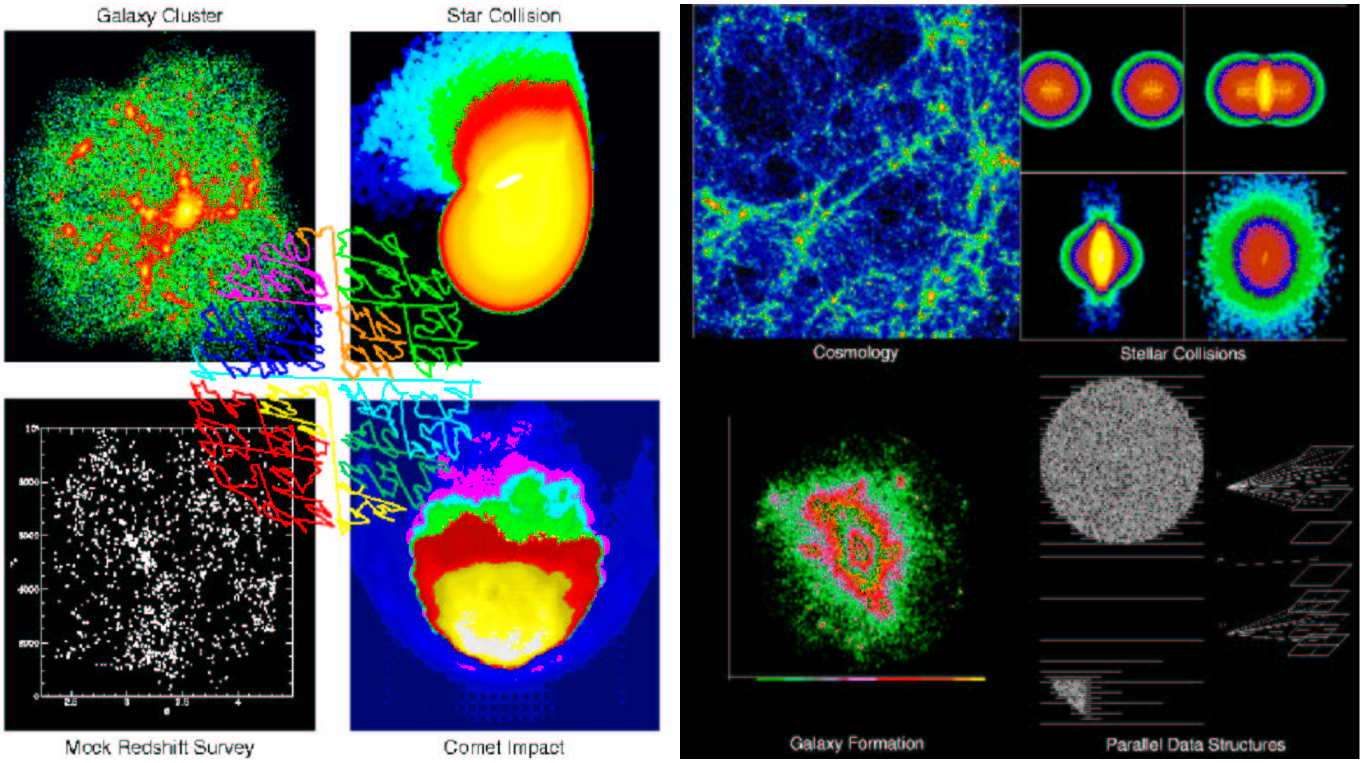

Figure 1.9: Several SPH applications in astrophysics realised at Los Alamos National Laboratory (*Theoretical Astrophysics* $(T-6)$).

[4] *http://qso.lanl.gov/pictures/Pictures.html*

SPH in solid mechanics

Nowadays, the range of applications of this method is much broader: in solid mechanics, it accurately simulates high distorsion and large deformation phenomena: for instance, linear and non linar oscillation of a plate have been successfully investigated by Gray [24] (see figure 1.10). This type of motion is quite delicate to model with finite element codes for instance. It is also successfully applied to solid breaking: on the left picture of figure 1.11, particles are used to model the ground as a penetrator passes through [5]. These kinds of simulations can predict the response of small components in the penetrator to a high-velocity impact. The picture on the right of figure 1.11 is a simulation of an airplane wing hitting a vertical pole. Fuel dispersal from the torn wing is accurately modeled with SPH particles [6].

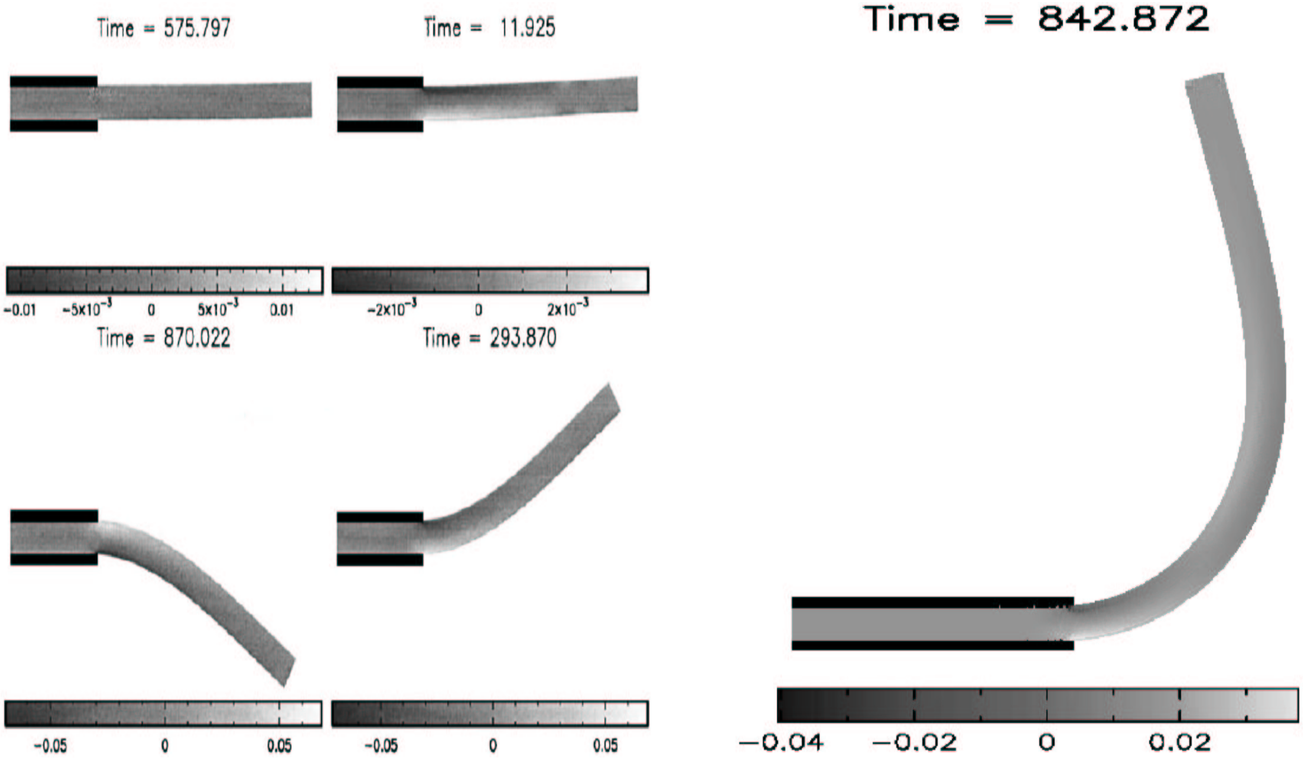

Figure 1.10: Linear (left) and non linear (right) oscillation of a plate modelled by SPH [24].

SPH in fluid mechanics

Being a particle based method, SPH is very well suited to model particles, such as bubbles entrained in flows. It is also appropriate for investigation of multiple phase mechanical engineering applications such as fluid jets, tank sloshing, multiple phase pipe flow, fluid impacts and cavitation.

[5] *http : //www.cs.sandia.gov/ sjplimp/pronto.html*

[6] *http : //www.cs.sandia.gov/ sjplimp/pronto.html*

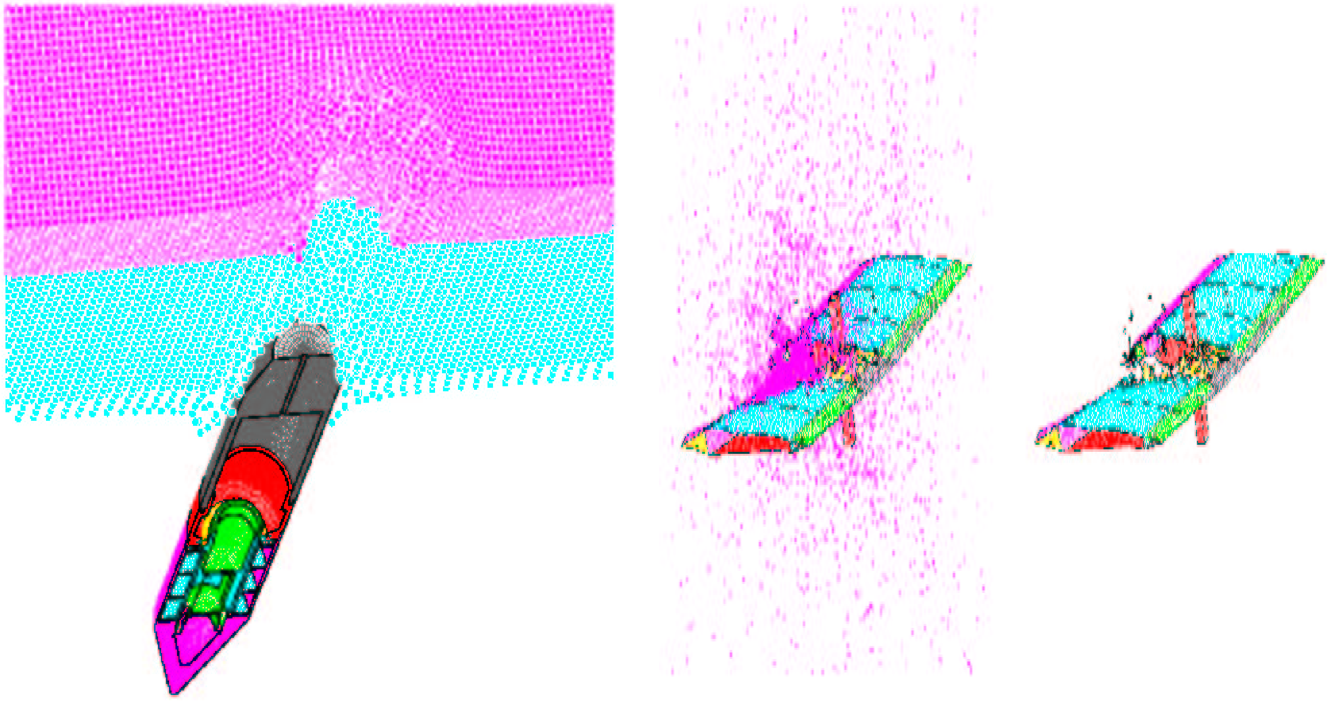

Figure 1.11: Several SPH applications in solid mechanics obtained by the code *Pronto*.

- Industrial applications

  High Pressure Die Casting (HPDC) is an important industrial process by which very complex shaped castings with excellent surface finishes can be produced in high volumes and at low cost. SPH is well suited to model momentum-dominated flows involving droplet formation and splashing, such as occur in HPDC. SPH predictions relative to a simple plate die have been compared with the corresponding experimental results of Schmid & Klein [26]: as seen in plots of figure 1.12, the SPH results compare well with the experimental results, both capture the essential nature of the flow. HPDC in a 2D channel is also well simulated by SPH, as shown in figure 1.13.

- Environmental applications

  SPH is also efficiently applied to "complex" environmental flows simulation. The term "complex" corresponds to delicate geometric conditions, such as chaotic free surface, hydraulic jumps and wave flumes. A dam breaking [91], [90] has been accurately represented by SPH with 10 000 particles, as shown in figure 1.14. One can notice that the breaking wave at $t = 1.20$ s is well modelled, while classical Eulerian codes struggle for these typical free surface problems. In the same way, a solitary wave on a beach is accurately simulated by SPH. The experimental pictures on the right of figure 1.15 reveal that there is an excellent agreement between experimental and numerical results

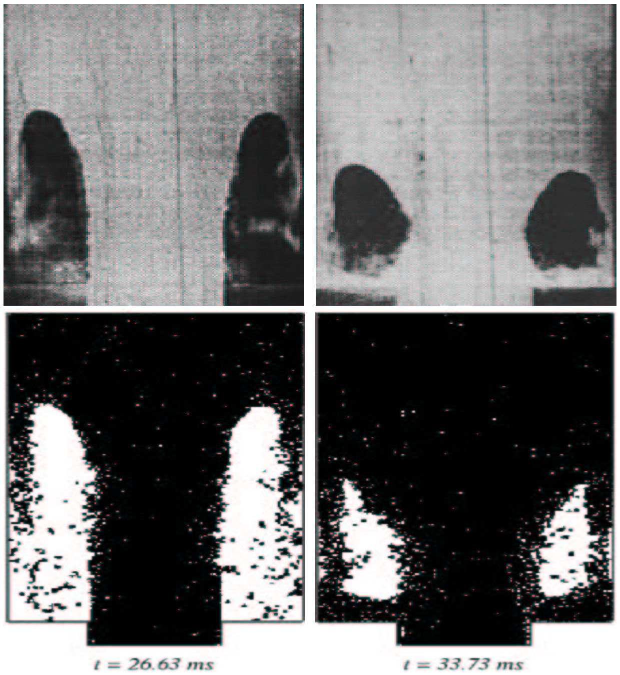

Figure 1.12: High Pressure Die Casting (HPDC) relative to a simple plate die realised by SPH (bottom pictures) and compared with experimental data (top pictures) [26].

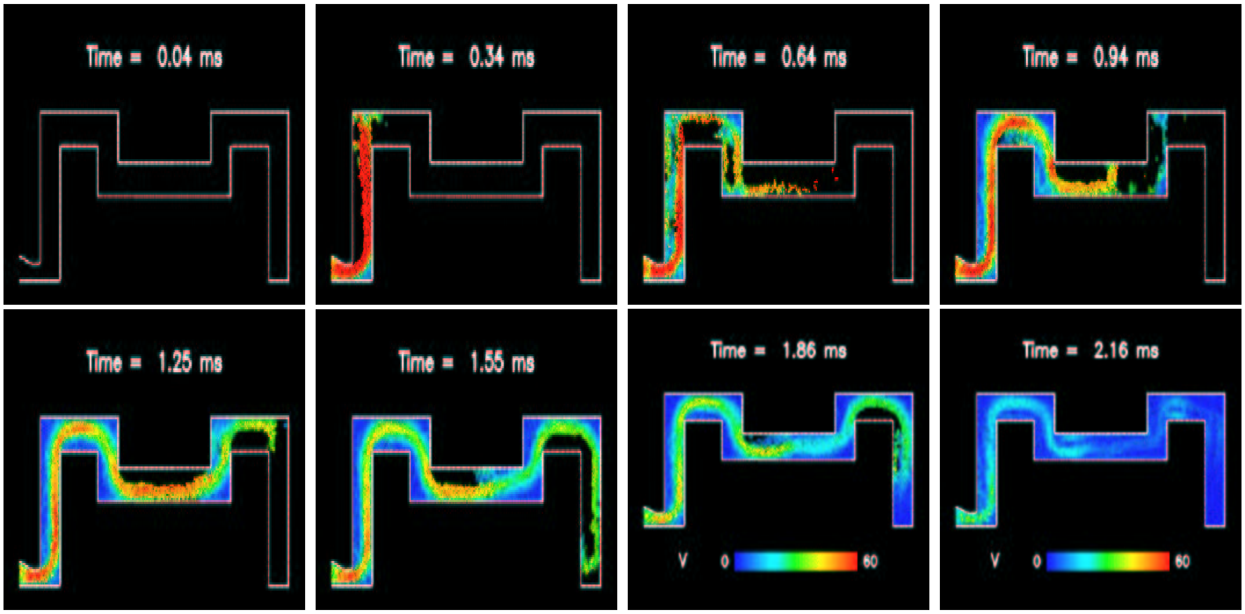

Figure 1.13: Velocity contours of the liquid metal during a channel HPDC realised by SPH [12].

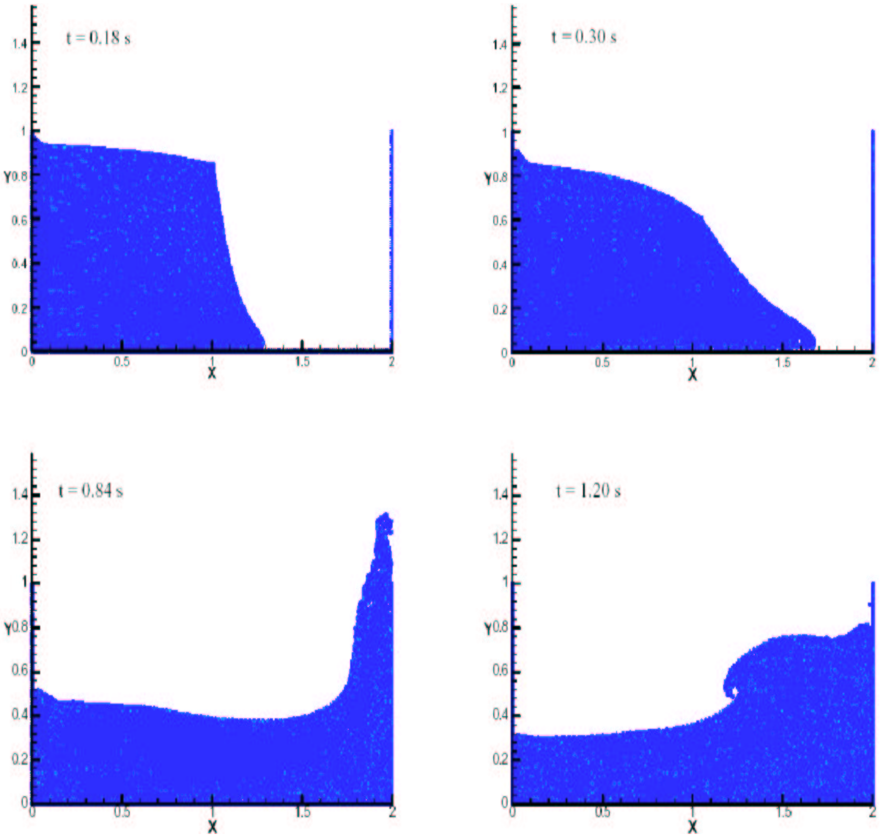

Figure 1.14: Dam breaking simulation with SPH [91].

relative to SPH.

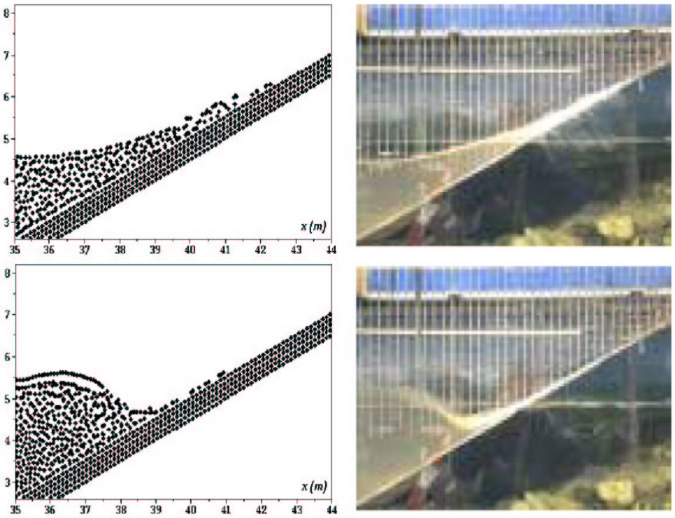

Figure 1.15: Solitary wave on a beach simulated with SPH (left pictures) and compared to experimental data (right pictures) [87].

1.3 Conclusion

Currently, several dozen particle and meshless methods exist. Each of them has its own advantages and drawbacks which make the method well suited for a specific type of problems. There is thus no "best method": the user needs to choose the appropriate method for the problem he has to solve. In order to simulate complex environmental flows, SPH seems to be an attractive method and natural way of separating water and air.

Chapter 2

The SPH method: bibliographic investigation

In this chapter, a theoretical and practical overview of SPH applied to CFD is presented.

2.1 Fluid discretisation in SPH formalism

2.1.1 SPH particle definition

In the physical space, the fluid is discretised by a finite number of macroscopic volumes of fluid. Each of these volumes v_a is itself composed of several mesoscopic fluid elements which are identical to the particles defined in continuum mechanics formalism (see figure 2.1). However, in SPH theory, the term “particle” always corresponds to the macroscopic volume of fluid v_a. In order to simplify the notations, the term “particle” will be written without any quotation mark in the following. In the three-dimensional Euclidian space, SPH particles are linked to a set of N points denoted by latin letters. Each point is located at a position $\underline{r}_a$ and corresponds in the physical space to the gravity centre of fluid particle a.

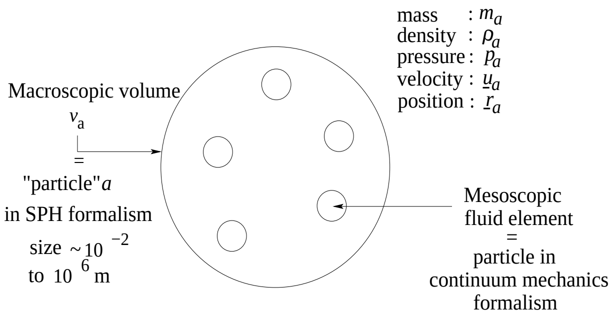

Figure 2.1: “Particle” definition in SPH formalism.

2.1.2 Physical values assigned to fluid particles

Each fluid particle a has a mass m_a, constant throughout the calculation. Particle a is also characterised by a density ρ_a, a pressure p_a and a velocity $\underline{u}_a$ (see figure 2.1), updated at each time step. Its pressure and velocity correspond to a statistical average relative to all mesoscopic fluid elements constituting particle a. It is the same for other intensive quantities. Concerning extensive quantities, they correspond to a summation relative to all mesoscopic fluid elements constituting the fluid particle a.

2.2 Mathematical formalism

2.2.1 Heart of the method

Let us consider a function A, which may represent a physical variable and is defined over a domain of interest Ω. In SPH formalism, the value of A at a point $\underline{r}$ is firstly written as a convolution product of the function A with the Dirac distribution δ (exact mathematical formulation):

$$A(\underline{r}) = \int_{\Omega} A(\underline{r}')\delta(\underline{r} - \underline{r}')dr' \tag{2.1}$$

where the summation is extended to the whole domain Ω.

For numerical reasons, the Dirac distribution is firstly approximated by a smooth function $w_h(\underline{r} - \underline{r}')$ called the kernel function. Further information regarding this function and the smoothing length h will be given in part 2.3. $A(\underline{r})$ can be approximated by the integral interpolant of the function A at the point $\underline{r}$, denoted $\langle A(\underline{r})\rangle$, according to

$$A(\underline{r}) \approx \langle A(\underline{r})\rangle = \int_{\Omega} A(\underline{r}')w_h(\underline{r} - \underline{r}')dr' \tag{2.2}$$

In order to determine the accuracy of equation (2.2), let $A(\underline{r}')$ be expanded in a Taylor series around $\underline{r}$, as decribed in [18]:

$$A(\underline{r}') = A(\underline{r}) + |\underline{r}' - \underline{r}|\dot{A}(\underline{r}) + O\left(|\underline{r} - \underline{r}'|^2\right) \tag{2.3}$$

where $\dot{A}(\underline{r})$ corresponds to the first derivative of A. Combining equations (2.2) and (2.3), it follows that

$$\langle A(\underline{r})\rangle = A(\underline{r}) \int_{\Omega} w_h\left(\underline{r} - \underline{r}'\right) dr' + \dot{A}(\underline{r}) \int_{\Omega} |\underline{r}' - \underline{r}| w_h\left(\underline{r} - \underline{r}'\right) dr' + O\left(|\underline{r} - \underline{r}'|^2\right) \tag{2.4}$$

If the kernel function w_h is spherical, i.e $w_h\left(\underline{r}-\underline{r}'\right)=w_h\left(|\underline{r}-\underline{r}'|\right)$, the term of order $O(|\underline{r}'-\underline{r}|)$ in equation (2.4)

$$\dot{A}\left(\underline{r}\right)\int_{\Omega}|\underline{r}'-\underline{r}|w_h\left(\underline{r}-\underline{r}'\right)dr' \tag{2.5}$$

automatically vanishes. Moreover, since the order of $|\underline{r}'-\underline{r}|$ is generally similar to the order of h (see part 2.3), equation (2.4) becomes:

$$\langle A\left(\underline{r}\right)\rangle=A\left(\underline{r}\right)\int_{\Omega}w_h\left(|\underline{r}-\underline{r}'|\right)dr'+O\left(h^2\right) \tag{2.6}$$

Since $\int_{\Omega}w_h\left(|\underline{r}-\underline{r}'|\right)dr'=1$ (see part 2.3), it follows that

$$\langle A\left(\underline{r}\right)\rangle=A\left(\underline{r}\right)+O\left(h^2\right) \tag{2.7}$$

Relation (2.7) gives a leading-error proportional to h^2 and results in an accuracy for the SPH discretisation which is in second-order in space [18]. Consequently,

$$A\left(\underline{r}\right)=\int_{\Omega}A\left(\underline{r}'\right)w_h\left(\underline{r}-\underline{r}'\right)dr'+O\left(h^2\right) \tag{2.8}$$

Statistical interpretation

As generally defined, the density ρ is proportional to the number of particles per unit volume. From a statistical point of view, the density ρ at a particle located at a spatial point $\underline{r}$ could be described by considering that the probability of finding a particle in a certain volume ΔV centred on $\underline{r}$ is proportional to $\rho\Delta V$. The position of the particles can thus be considered as a sample from a probability density proportional to the density [23]. By considering the function A of equation (2.8) as the density ρ, it is clear that the density of the particle located at $\underline{r}$ is proportional to the density probability to find a particle in a volume element centred on $\underline{r}$. This probability density is here equal to $\rho\left(\underline{r}'\right)w_h\left(\underline{r}-\underline{r}'\right)$. SPH relation (2.8) is hence consistent with the statistical interpretation of the density.

2.2.2 Transition to a discrete domain

The transition to a discrete domain is achieved by approximating the integral of equation (2.8) by a Riemann summation:

$$A(\underline{r})=\sum_b\frac{m_b}{\rho_b}A_b w_h(\underline{r}-\underline{r}_b)+O(h^2) \tag{2.9}$$

where A_b denotes the value of A at the point occupied by the particle b. This summation is extended to all particles b that constitute the fluid domain (the volume element dr' has been

replaced by the particle volume m_b/ρ_b). An approximation of equation (2.8) by the statistical Monte Carlo method applied to a set of N points also leads to relation (2.9) [23].
With a spherical kernel, the value of the quantity A relative to the particle a located at the point $\underline{r}$ can then be written in the following form

$$A(\underline{r}_a) = A_a = \sum_b \frac{m_b}{\rho_b} A_b w_h(r_{ab}) + O\left(h^2\right) \tag{2.10}$$

where the quantity r_{ab} corresponds to the distance between particles a and b. In order to simplify expressions, the order of approximation $O\left(h^2\right)$ will be skipped in the following. Moreover, only spherical kernels will be considered.

"scatter" and "gather" point of view

There are two classical interpretations of equation (2.10):

- The "scatter" interpretation
  In this interpretation, the kernel function w_h can be seen as a tool to describe how the thermodynamic variables for a particle b are distributed at the position $\underline{r}_a$ occupied by the particle a. From this point of view, the kernel function is thus centered on particle b, as shown on the left picture of figure 2.2. On this figure, the kernel is represented as a Gaussian function.

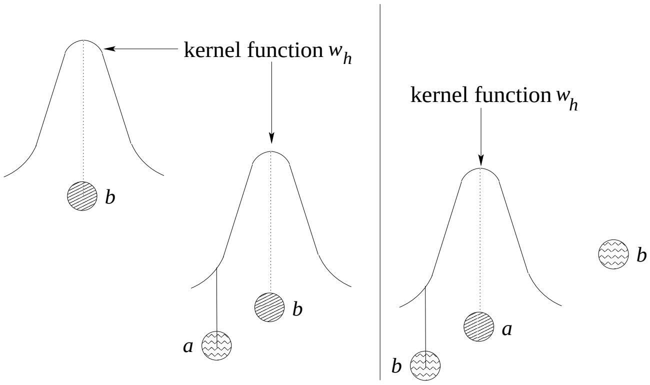

Figure 2.2: The "scatter" (left) and "gather" (right) interpretation of SPH.

- On the other hand, the "gather" interpretation considers the kernel function as a weighting function used to describe how thermodynamic variables at the particle a are smoothed in its neighbourhood. In this second interpretation, the kernel function is centered on the particle a, as represented on the right picture of figure 2.2.

The denomination *Smoothed Particle Hydrodynamics* can be understood from these two points of view: the quantities relative to each particle are smoothly smeared over a finite space region.

<u>*Remarks:*</u>

- At this point, one can notice from equation (2.10) that the contribution of particle b to the value of the quantity A relative to particle a directly depends upon the distance between these two particles.
- Moreover, equation (2.10) reveals that all particles present in the domain are involved in the determination of the value of the quantity A relative to particle a, which is very time consuming. To avoid this drawback, it is convenient to consider a kernel characterised by a compact support (see part 2.3).

2.2.3 Gradient of a scalar field

Basic formulation

The interpolant of the function A established according to equation (2.10) is differentiable provided the kernel function is also differentiable. Then, the gradient of a scalar field A relative to the particle a can be written

$$(\nabla A)_a = \sum_b \frac{m_b}{\rho_b} A_b \nabla_a w_h(r_{ab}) = \sum_b \frac{m_b}{\rho_b} A_b \dot{w}_h(r_{ab}) \underline{\epsilon}_{ab} \tag{2.11}$$

the quantity $\nabla_a w_h\,(r_{ab})$ denotes the gradient of the kernel taken with respect to a-coordinates and $\dot{w}_h$ the kernel derivative. $\underline{\epsilon}_{ab}$ corresponds to the vector $\underline{r}_{ab}/r_{ab}$, where $\underline{r}_{ab} = \underline{r}_a - \underline{r}_b$, as represented on figure 2.3. One can notice that there is no need to use a grid to evaluate the gradient of a scalar field since it is a function of the kernel gradient which is analytically known (see part 2.3). This SPH feature is essential and very attractive.

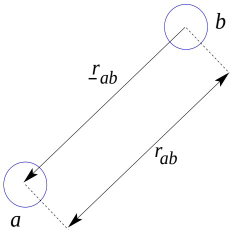

Figure 2.3: Definition of the particle vector $\underline{r}_{ab}$.

Expression (2.11) is not an accurate one since a constant function A will not automatically give a zero gradient function. Fortunately, as in finite-differences methods, the gradient of a scalar field A can be written in several ways in SPH formalism [56].

Other formulations

By considering relations (2.11) and (2.12),

$$\rho\nabla A = \nabla(\rho A) - A\nabla\rho \tag{2.12}$$

one obtains:

$$(\nabla A)_a = \frac{1}{\rho_a}\sum_b m_b(A_b - A_a)\dot{w}_h(r_{ab})\underline{\epsilon}_{ab} \tag{2.13}$$

In this case, the gradient of a constant function A is obviously zero. Moreover, it is shown on figure 2.9 that this expression is more accurate than equation (2.11).
In the same way and considering the following relation

$$\nabla A = \rho\nabla\left(\frac{A}{\rho}\right) + \frac{1}{\rho}A\nabla\rho \tag{2.14}$$

another form of the gradient of a scalar field is obtained:

$$(\nabla A)_a = \rho_a\sum_b m_b\left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2}\right)\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} \tag{2.15}$$

Even if it is not clear that equation (2.15) gives a zero gradient for a constant function A, numerical tests reveal that this last approximation is as accurate as equation (2.13) [23]. For variational consistency reasons, Bonet [6] established also the following form

$$(\nabla A)_a = \sum_b \frac{m_b}{\rho_b}\left(A_a + A_b\right)\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} \tag{2.16}$$

Monaghan presents several forms of scalar field gradients and discusses their advantages and drawbacks in [56].

Symmetric and asymmetric consideration

Equation (2.13) is symmetric with respect to a and b subscripts. Indeed, the kernel derivative is anti-symmetric with respect to a and b subscripts, i.e.

$$\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} = -\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ba} \tag{2.17}$$

Consequently, the contribution of particle b to $\left(\nabla A\right)_a$ is identical to the contribution of particle a to $\left(\nabla A\right)_b$. In the same way, it is easy to show that equations (2.15) and (2.16) are asymmetric with respect to a and b subscripts. From a physical point of view, these considerations are very important (see part 2.4) and from a numerical point of view, calculation time is spared by using them: there is no need to calculate the contribution of particle a to $(\nabla A)_b$ if the contribution of particle b to $(\nabla A)_a$ is already computed.

2.2.4 Divergence of a vector field

Several formulations

By considering expressions such as equations (2.12) and (2.14), several forms of the divergence of a vector field can be written in SPH formalism. Among them,

$$(\nabla.\underline{A})_a = \frac{1}{\rho_a}\left[\nabla.\left(\rho\underline{A}\right)\right]_a - \frac{1}{\rho_a}\underline{A}_a.\left(\nabla\rho\right)_a = \frac{1}{\rho_a}\sum_b m_b\left(\underline{A}_b - \underline{A}_a\right)\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} \tag{2.18}$$

$$(\nabla.\underline{A})_a = \rho_a\left[\nabla.\left(\frac{1}{\rho}\underline{A}\right)\right]_a + \frac{1}{\rho_a}\underline{A}_a.\left(\nabla\rho\right)_a = \rho_a\sum_b m_b\left(\frac{\underline{A}_a}{\rho_a^2} + \frac{\underline{A}_b}{\rho_b^2}\right)\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} \tag{2.19}$$

As the gradient operator, the divergence one can be written in a symmetric (equation 2.18) and an asymmetric (equation 2.19) form with respect to a and b subscripts.

Velocity divergence

If the field A corresponds to the velocity $\underline{u}$, the physical interpretation of equation (2.18) is quite simple by considering a Gaussian kernel, defined in 1D by

$$w_h\left(x\right) = \frac{1}{h\sqrt{\pi}}\exp\left(-\frac{x^2}{h^2}\right) \tag{2.20}$$

and represented on the right of figure 2.7. According to equations (2.18) and (2.20), the contribution of particle b to the divergence of the velocity at particle a is

$$\frac{2m_b}{\rho_a}\underline{u}_{ab}.\underline{r}_{ab}\frac{w_h\left(r_{ab}\right)}{h^2} \tag{2.21}$$

where $\underline{u}_{ab} = \underline{u}_a - \underline{u}_b$ and $\underline{r}_{ab} = \underline{r}_a - \underline{r}_b$. Relation (2.21) reveals that this contribution is positive (respectively negative) if particles a and b move away from (resp. get closer to) each other, which is consistent with the classical interpretation of the velocity divergence. This interpretation is illustrated on figure 2.4.

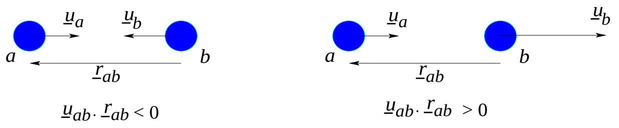

Figure 2.4: Interpretation of the velocity divergence in SPH formalism.

2.2.5 Vector gradient and tensor divergence

Previous equations still hold for operators with higher dimensions. One can hence write the gradient of a vector $\underline{A}$ in the following forms:

$$\left(\underline{\underline{\nabla}}.\underline{A}\right)_a = \frac{1}{\rho_a} \sum_b m_b \left(\underline{A}_b - \underline{A}_a\right) \otimes \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.22}$$

$$\left(\underline{\underline{\nabla}}.\underline{A}\right)_a = \rho_a \sum_b m_b \left(\frac{\underline{A}_a}{\rho_a^2} + \frac{\underline{A}_b}{\rho_b^2}\right) \otimes \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.23}$$

where ⊗ defines a tensor product. In the same way, the divergence of a second order tensor $\underline{\underline{A}}$ can be written in SPH formalism according to

$$\left(\underline{\underline{\nabla}}.\underline{\underline{A}}\right)_a = \frac{1}{\rho_a} \sum_b m_b \left(\underline{\underline{A}}_b - \underline{\underline{A}}_a\right) .\dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.24}$$

$$\left(\underline{\underline{\nabla}}.\underline{\underline{A}}\right)_a = \rho_a \sum_b m_b \left(\frac{\underline{\underline{A}}_a}{\rho_a^2} + \frac{\underline{\underline{A}}_b}{\rho_b^2}\right) .\dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.25}$$

As previously mentionned, each operator has a symmetric (equations (2.22) and (2.24)) and a asymmetric form (equations (2.23) and (2.25)).

2.3 The kernel function

2.3.1 Fundamental properties

Let us consider a domain of interest Ω. The kernel function w_h must satisfy the three following properties:

- The kernel summation over the whole domain Ω must be equal to unity, i.e

$$\int_\Omega w_h \left(\underline{r} - \underline{r}'\right) dr' = 1 \tag{2.26}$$

  This condition ensures proper normalization [18].

- When the parameter h tends to zero (this parameter called the smoothing length will be developed in part 2.3.6), the kernel function must tend to the Dirac distribution:

$$\lim_{h \to 0} w_h \left(\underline{r} - \underline{r}'\right) = \delta \left(\underline{r} - \underline{r}'\right) \tag{2.27}$$

  This feature ensures consistency in the continuum limit.

- The kernel must be at least once differentiable and its derivative must be continuous in order to avoid large fluctuations which would affect the solution [23].

2.3.2 Compact support

In order to spare calculation time, it is useful to consider a kernel w_h characterised by a compact support: $w_h(r_{ab})$ is then zero if the distance r_{ab} between two particles exceeds a limit value equal to the kernel support size h_t. The summation of equation (2.10) will thus be restricted to particles b which are considered as the nearest neighbours of particle a: as shown on figure 2.5, only the particles located in the disc (in the sphere in 3D) of radius h_t and centred on a will be involved in equation (2.10). Concerning the kernel support size h_t, it is directly proportional to the smoothing length h.

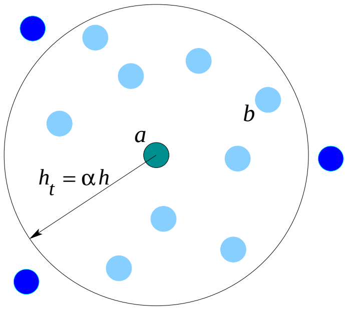

Figure 2.5: Neighbours of particle a with a compact support kernel.

2.3.3 Examples of kernel

In a general way, a kernel function can be written according to:

$$w_h(r_{ab}) = \frac{1}{h^\sigma} f\left(\frac{r_{ab}}{h}\right) \tag{2.28}$$

where σ denotes the dimension of the problem. Concerning the function f, several formulations exist: for a 2D problem, Monaghan and Lattanzio [56] have defined the third order spline kernel with

$$f(q) = \frac{10}{7\pi} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if } 0 \leq q \leq 1 \\ \frac{1}{4}(2-q)^3 & \text{if } 1 \leq q \leq 2 \\ 0 & \text{if } q \geq 2 \end{cases} \tag{2.29}$$

where q corresponds to the ratio r_{ab}/h. In this case, the kernel support size h_t is equal to $2h$. In the same way, a fourth order spline kernel is defined with

$$f\left(q\right)=\frac{51}{2000}\begin{cases}\left(\frac{5}{2}-q\right)^4-5\left(\frac{3}{2}-q\right)^4+10\left(\frac{1}{2}-q\right)^4 & \text{if } 0\leq q\leq 0.5\\ \left(\frac{5}{2}-q\right)^4-5\left(\frac{3}{2}-q\right)^4 & \text{if } 0.5\leq q\leq 1.5\\ \left(\frac{5}{2}-q\right)^4 & \text{if } 1.5\leq q\leq 2.5\\ 0 & \text{if } q\geq 2.5\end{cases} \tag{2.30}$$

The kernel support size h_t is here equal to $2.5h$.
In order to investigate a low Reynolds number flow, Morris [64] used a fifth order spline kernel defined with

$$f\left(q\right)=\frac{7}{478\pi}\begin{cases}\left(3-q\right)^5-6\left(2-q\right)^5+15\left(1-q\right)^5 & \text{if } 0\leq q\leq 1\\ \left(3-q\right)^5-6\left(2-q\right)^5 & \text{if } 1\leq q\leq 2\\ \left(3-q\right)^5 & \text{if } 2\leq q\leq 3\\ 0 & \text{if } q\geq 3\end{cases} \tag{2.31}$$

The third, fourth and fifth order spline kernels are represented on figures 2.6 and 2.7 with $x_{ab}=x_a-x_b$ and $z_{ab}=z_a-z_b$: they resemble a Gaussian function (see the right picture of figure 2.7) but are characterised by a compact support. Moreover, their first (left of figure 2.8) and second (right of figure 2.8) derivative are continuous, which implies that the interpolation is not too sensitive to particle disorder [18]. Using different kernels in SPH is similar to using different numerical schemes in finite-differences and finite volumes codes. Indeed, a numerical

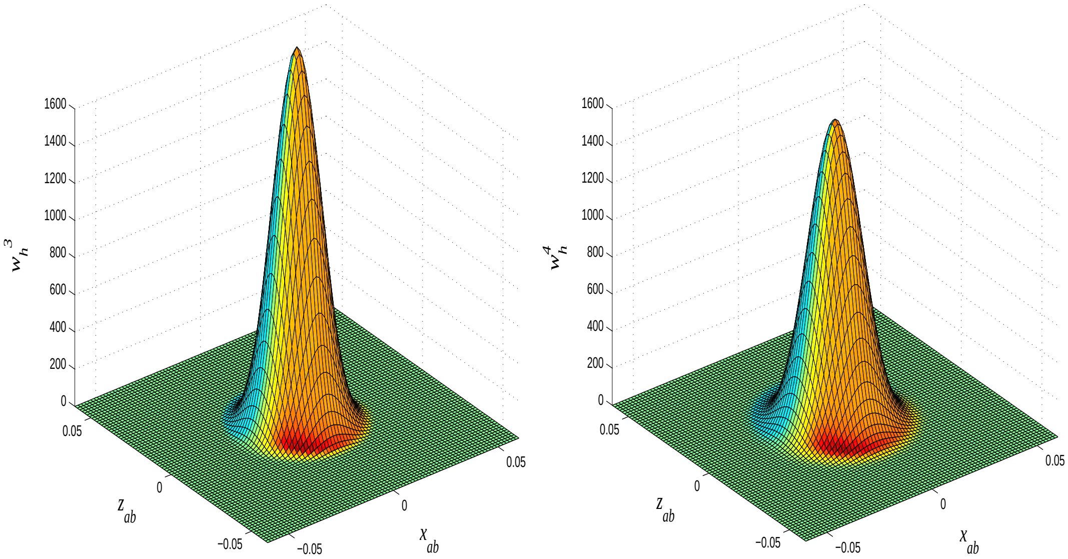

Figure 2.6: Third (left) and fourth (right) order spline kernels.

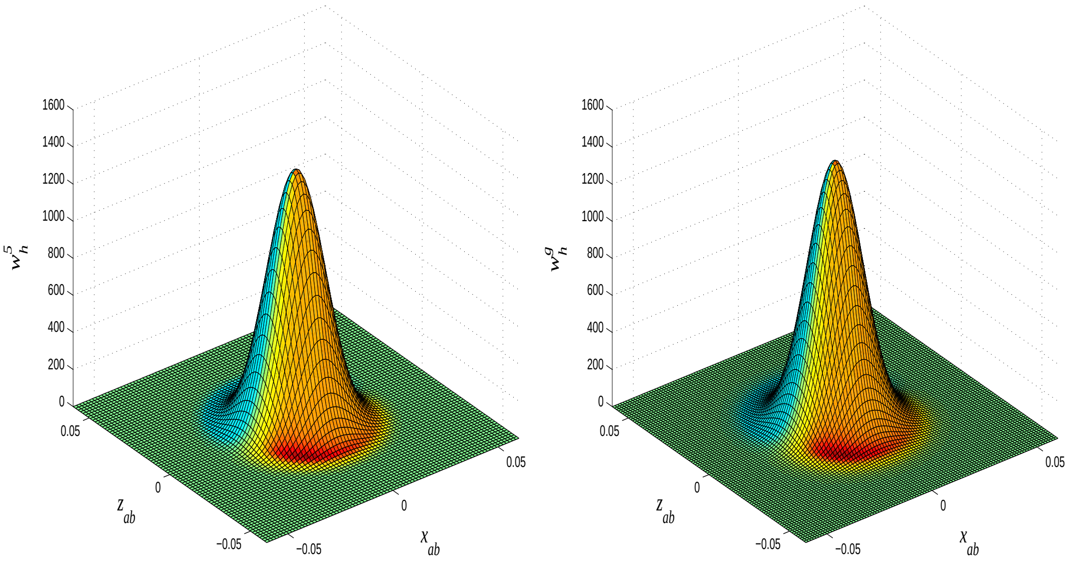

Figure 2.7: Fifth order spline (left) and Gaussian kernels (right).

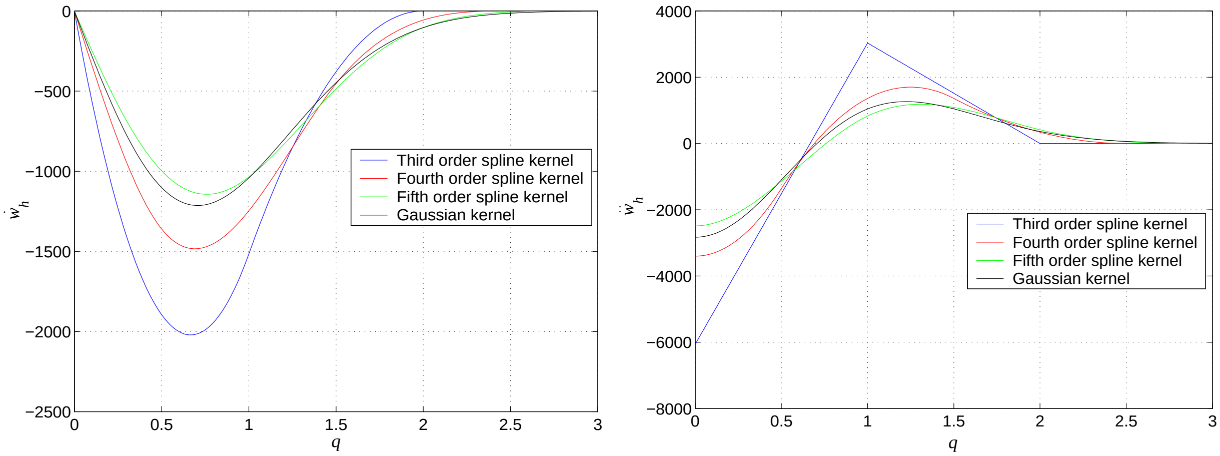

Figure 2.8: First (left) and second (right) derivatives of the third order spline (blue), fourth order spline (red), fifth order spline (green) and Gaussian (black) kernels.

scheme determines the connections relative to a node, as the kernel function does. Moreover, both are linked to the accuracy of the equation discretisation. In the finite element method, the kernel function is similar to the shape function. Different equations can also have different kernels, as different schemes can be used to discretise an equation set with finite-differences methods.

2.3.4 Kernel accuracy and stability

Each kernel is characterised by specific accuracy and stability properties.

- The order of spline kernels determines their approximation order. Indeed, let us consider the basic SPH relation (2.2)

$$\langle A\left(\underline{r}\right)\rangle = \int_{\Omega} A\left(\underline{r}'\right) w_h\left(\underline{r}-\underline{r}'\right) d\underline{r}' \tag{2.32}$$

  Considering a simple 1D case, the function $A\left(\underline{r}'\right)$ can be approximated by a Taylor series expansion about $\underline{r}$, as explained in part 2.2.1:

$$\langle A\left(r\right)\rangle = A\left(r\right)\int_{\Omega} w_h\left(r-r'\right) dr' + \dot{A}\left(r\right)\int_{\Omega}\left(r'-r\right) w_h\left(r-r'\right) dr' + \dots \tag{2.33}$$

  The condition

$$\int_{\Omega}\left(r'-r\right)^j w_h\left(r-r'\right) dr' = 0 \qquad 0 < j \leq k \tag{2.34}$$

  is achieved by considering a k-order spline kernel and $\langle A\left(r\right)\rangle$ will then coincide with $A\left(r\right)$.

- In [24], Gray established a SPH dispersion relation by stacking a small perturbation to the equation of motion. By comparing this dispersion relation to the exact one relative to a continuum medium, he revealed that a long-wavelength approximation of the SPH dispersion relation exactly gives the continuum equation. This firstly shows that the SPH perturbation equations are consistent in the long-wavelength limit. With a short wavelength approximation, the SPH dispersion relation involves the Fourier transform of the kernel. The accuracy of a kernel is thus linked to its Fourier transform. Indeed, Monaghan [56] showed that the more accurate the kernels are, the faster their Fourier transform decreases with the wave number. For instance, he considered a one-dimensional case where particles of equal mass are equispaced by a distance λ. He proved that the density ρ_a for a particle a estimated in SPH formalism, can be written according to

$$\rho_a = \frac{m}{v}\left[1 + F\left(\frac{2\pi h}{\lambda}\right)\right] \tag{2.35}$$

where F denotes the Fourier transform of the kernel and v the particle volume. By considering the Gaussian kernel (defined by equation (2.20)), one obtains:

$$F\left(\frac{2\pi h}{\lambda}\right) = \sqrt{\pi}\exp\left(-\frac{4\pi^4 h^2}{\lambda^2}\right) \tag{2.36}$$

Considering $(h/\lambda) = 1.5$, equation (2.35) becomes

$$\rho_a = \frac{m}{v} \tag{2.37}$$

- Kernel accuracy could be evaluated as well by estimating their capability to reproduce a known density distribution [23].

Concerning kernel stability, Morris [64] revealed that SPH equations can be unstable when kernels with compact support are used. As higher order spline kernels more closely approximating a Gaussian are used, these instabilites are reduced. One reason for the poor performance of lower order splines kernels is that the stability properties of SPH equations strongly depends upon the second derivative of the kernel. This consideration will be developed further in part 2.8.1.

2.3.5 Kernel correction

Bonet [6] presents a kernel correction, developed by Liu, in order to ensure that polynomial functions up to a given degree are exactly interpolated. Indeed, due to the point interpolation of equation (2.10), the kernel interpolation does not give perfect results. This can be improved by interpolating with a corrected kernel $\tilde{w_h}$ according to

$$A_a = \sum_b \frac{m_b}{\rho_b} A_b \tilde{w_h}\left(r_{ab}\right) \tag{2.38}$$

where for a linear correction $\tilde{w_h}$ is defined as

$$\tilde{w_h}(r_{ab}) = w_h(r_{ab})\alpha(r_{ab})\left[1 + \beta(r_{ab})r_{ab}\right] \tag{2.39}$$

The parameters $\alpha\left(r_{ab}\right)$ and $\beta\left(r_{ab}\right)$ are evaluated by enforcing that any linear distribution for instance is exactly interpolated:

$$\underline{v}_0 + v_1.\underline{r}_a = \sum_b \frac{m_b}{\rho_b}\left(\underline{v}_0 + v_1.\underline{r}_b\right)\tilde{w_h}\left(r_{ab}\right) \tag{2.40}$$

where $\underline{v}_0$ and v_1 respectively correspond to arbitrary vector and scalar. The use of this type of correction ensures that linear functions are perfectly interpolated and their gradients are exactly obtained. However, the evaluation of the parameters α and β is quite cumbersome.

Moreover, the computation of the gradient of A is much more time consuming. A possible way of simplifying the calculation is by using constant, rather than linear correction. This is equivalent to considering $\beta = 0$ in equation (2.39) and gives

$$A_a = \frac{\sum_b \frac{m_b}{\rho_b} A_b w_b (r_{ab})}{\sum_b w_b (r_{ab})} \tag{2.41}$$

Equations (2.38) and (2.41) are efficiently tested by Bonet in [6]. Both provide a much improved interpolation for the gradient of a velocity field but they still remain quite time consuming.

2.3.6 The smoothing length h

In SPH theory, the smoothing length h characterises the spatial discretisation of the problem and is analogous to an average cell size in an Eulerian code. This parameter, linked to the kernel support, strongly influences result accuracy.

h optimal

Gingold and Monaghan [23] have estimated kernel accuracy by testing their capability to reproduce known density distributions. By considering the error obtained on the computed density, they revealed the existence of an optimal h value for each particle number, as shown on figure 2.9. This is consistent with the work of Speith [76] who proved that the accuracy of SPH results depends on the number of neighbours N_i associated with each particle: Speith

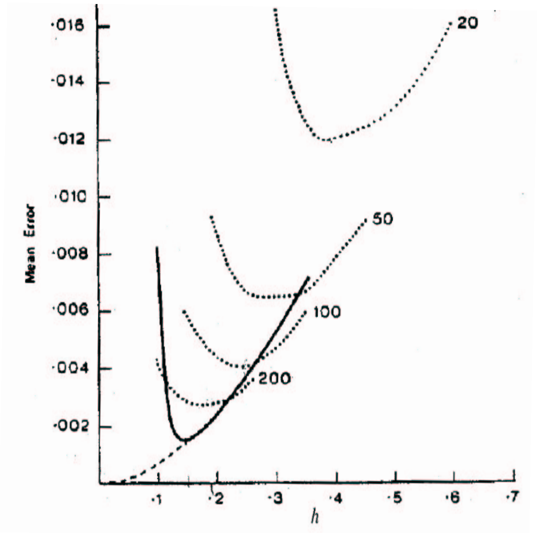

Figure 2.9: Mean error relative to a density evaluation simulation. 20, 50, 100 and 200 particles are successively considered.— — : pressure evaluated according to equation (2.11) ; —— : pressure evaluated according to equation (2.13) with 100 particles [23].

explained that

$$N_i \propto h^d N \tag{2.42}$$

where N corresponds to the total number of particles and d the dimension of the problem. According to equation (2.42), a high value of h results in a large kernel support and thus increases the number of neighbours but numerical errors associated with equation (2.10) are also more consequent. Moreover, physical variations will also be smoothed out if a too large smoothing length is used. On the other hand, the number of neighbours will be too low if the smoothing length is adjusted to the smallest characteristic length scale of the problem. An optimal value of h for a given particle number has consequently to be found. Once this value is obtained, the ratio $h/\delta r$ (where δr corresponds to the initial interparticle spacing) is constant (see chapter 4).

Specific h

By considering a specific h for each particle, results for a density evaluation simulation (see figure 2.10) are more satisfactory: in this case, the maximal error is six times lower than with a constant smoothing length. However, it results in more complex equations (see part 2.4).

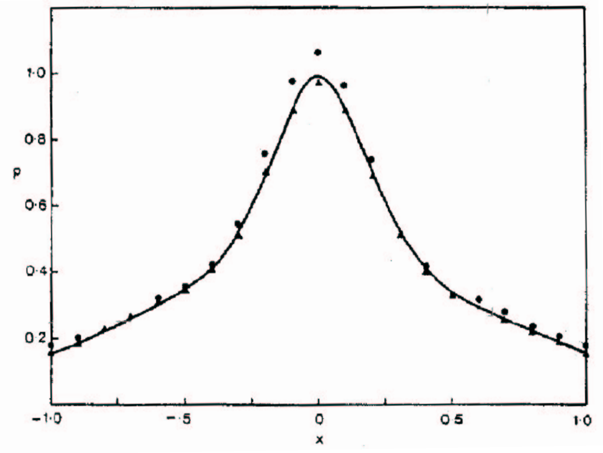

Figure 2.10: Density estimation with a Gaussian kernel. —— : theoretical density, △: specific h, • : constant h [23].

Moreover, at each time step, h value must be updated for each particle, hence increasing the calculation time.

2.4 Fluid mechanics equations in SPH formalism

In this work, the fluid description is not energy-dependent. Therefore, no energy equation will be presented, nor will energy conservation question be considered. The reader can find information about these last points in [56], [61].

2.4.1 The continuity equation

Classical forms

- A natural formulation of the continuity equation can be deduced from equation (2.10), with A corresponding to the density ρ:

$$\rho_a = \sum_b m_b w_h \left(r_{ab}\right) \tag{2.43}$$

- In classical Lagrangian formalism, the continuity equation is written:

$$\frac{d\rho}{dt} = -\rho \nabla \underline{u} = -\nabla \left(\rho \underline{u}\right) + \nabla \rho . \underline{u} \tag{2.44}$$

  where the operator d/dt denotes a Lagrangian derivative and $\underline{u}$ the velocity field. Using relation (2.11), a SPH form of the continuity equation (2.44) can be deduced:

$$\frac{d\rho_a}{dt} = \sum_b m_b \underline{u}_{ab} \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.45}$$

  with $\underline{u}_{ab} = \underline{u}_a - \underline{u}_b$.

In a general way, fluid mechanics equations written in SPH formalism have a simple physical interpretation. According to equation (2.45) and by considering a Gaussian kernel (see equation (2.20)), the contribution of particle b to the density evolution of particle a is equal to

$$2 m_b \underline{u}_{ab} \cdot \underline{r}_{ba} \frac{w_h \left(r_{ab}\right)}{h^2} \tag{2.46}$$

Relation (2.46) reveals that a-density increases (respectively decreases) when particle b gets closer (resp. moves appart) to particle a, which is physically meaningful.

Comparison of the two classical formulations

- From a compressible point of view, the mass of the system is conserved with formulations (2.43) and (2.45), provided the number of particles is constant. Indeed, the mass of each particle is constant throughout the calculation. However, from an incompressible point of view, it seems that both formulations do not exactly conserve the system mass since the velocity divergence is not identically zero.

- When simulating incompressible free surface flows ([57], [64]), expression (2.43) results in a density underestimation in the vicinity of the free surface. Indeed, in this zone the kernel is truncated (its summation is no more equal to unity) since particles suffer from a lack of neighbours, as shown for particle a on figure 2.11. Moreover, according to the state equation used (see part 2.4.3), a slight density variation induces an important pressure variation which degrades calculations. Numerical tests revealed that expression (2.45) reduces density estimation errors in the vicinity of a free surface ([57], [56], [59]).

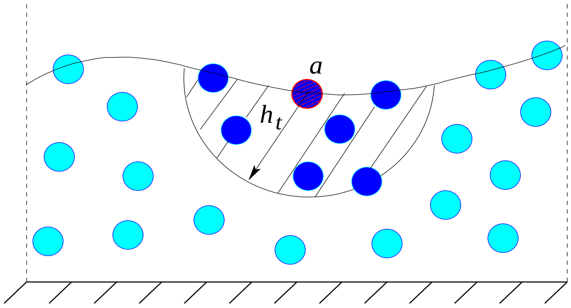

Figure 2.11: Neighbours lack in the vicinity of a free surface.

- Moreover, as equation (2.45), the momentum equation written in SPH formalism (see part 2.4.2) involves the kernel gradient and not the kernel values. From a numerical point of view, expression (2.45) is thus more attractive since the kernel gradient can be calculated in one subroutine and used for both the continuity and the momentum equations.

2.4.2 The equation of motion

Newton's second law relative to a particle a is written

$$m_a \underline{a}_a = \underline{F}_a^e - \underline{T}_a \tag{2.47}$$

where $\underline{F}_a^e$ represents the external forces acting on a such as self weight or Lorentz force in MHD, and $\underline{T}_a$ denotes the equivalent internal forces due to the state of stress inside the material. $\underline{a}_a$ corresponds to the a-acceleration. The internal forces $\underline{T}_a$ are typically evaluated from the Cauchy's equation $\rho_a \underline{a}_a = \frac{\rho_a}{m_a} \underline{F}_a^e + div\ \underline{\sigma}_a$. For an incompressible Newtonian fluid, the components σ_{ij} of the stress tensor $\underline{\sigma}$ are approximated by:

$$\sigma_{ij} = -p\delta_{ij} + 2\mu s_{ij} \tag{2.48}$$

where p corresponds to the dynamic pressure and μ the dynamic viscosity of the fluid. The deviatoric stress tensor components s_{ij} are defined by

$$s_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.49}$$

Combining equations (2.47), (2.48) and (2.49), one obtains the Lagrangian form of the momentum equation for a particle a:

$$\frac{d\underline{u}_a}{dt} = -\frac{1}{\rho_a}\nabla p_a + \nu \Delta \underline{u}_a + \underline{F}^e_a \tag{2.50}$$

d/dt corresponds to a Lagrangian derivative (as mentioned for equation (2.45)) and ν denotes the kinematic viscosity of the fluid.

Pressure gradient in SPH formalism

As explained in part 2.2.3, the pressure gradient term relative to particle a can be written in different forms in SPH formalism. The most widespread ones are:

$$\frac{1}{\rho_a}\nabla p_a = \sum_b \frac{m_b}{\rho_b \rho_a}\left(p_b + p_a\right) \dot{w}_h\left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.51}$$

and

$$\frac{1}{\rho_a}\nabla p_a = \sum_b m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2}\right) \dot{w}_h\left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.52}$$

The previous two expressions are equally accurate and anti-symmetric with respect to a and b subscripts. Considering a Gaussian kernel and equation (2.52), it can be shown that the contribution of particle b to the pressure gradient term relative to particle a is:

$$2m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2}\right) \frac{w_h\left(r_{ab}\right)}{h^2} \underline{r}_{ab} \tag{2.53}$$

which represents a momentum flux from b to a. Consequently, the pressure gradient term written in SPH formalism corresponds to a central repulsive force between particle pairs, as intermolecular forces between atoms.

Some SPH forms of the pressure gradient term (and not all of them) can be derived from a variational principle [6], [84]. For that, one firstly needs to define the total stored energy functional for the system. The variation of this functional leads to the equivalent internal forces relative to a particle ($\underline{T}_a$ in equation (2.47)). Bonet showed by this method that expression (2.52) (respectively (2.51)) is variationally consistent with the continuity equation (2.43) (resp. (2.45)). He also revealed that mixing the density equation and the pressure gradient term in an inconsistent way leads to poor results [6].

Conservation of linear and angular momenta

The total linear momentum $\underline{P}$ of a system of N particles is given by:

$$\underline{P} = \sum_{a=1}^{N} m_a \underline{u}_a \tag{2.54}$$

Combining the time derivative of this equation with the Newton's second law in the absence of external forces, the rate of change of the total linear momentum is given by:

$$\underline{\dot{P}} = \sum_{a=1}^{N} m_a \underline{a}_a = -\sum_{a=1}^{N} \underline{T}_a \tag{2.55}$$

Therefore, the condition for preservation of linear momentum is

$$\sum_{a=1}^{N} \underline{T}_a = \underline{0} \tag{2.56}$$

for any stress distribution. It is quite easy to prove whether a SPH relation preserves or not linear momentum. The internal forces $\underline{T}_a$ relative to a particle a can be expressed as the sum of interaction forces $\underline{T}_{ab}$ between pairs of particles as:

$$\underline{T}_a = \sum_{b} \underline{T}_{ab} \tag{2.57}$$

Considering equation (2.50) and neglecting viscous effects, it follows that

$$\underline{T}_a = \frac{m_a}{\rho_a} \nabla p_a \tag{2.58}$$

Equation (2.51) thus gives

$$\underline{T}_{ab} = \frac{m_a m_b}{\rho_b} \left(p_b + p_a\right) \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{2.59}$$

Since $\underline{\epsilon}_{ab} = -\underline{\epsilon}_{ba}$, it directly comes that $\underline{T}_{ab} = -\underline{T}_{ba}$. The total sum of all interaction pairs will thus vanish. The derivation for the linear momentum conservation of equation (2.52) is identical. The property $\dot{w}_h \underline{\epsilon}_{ab} = -\dot{w}_h \underline{\epsilon}_{ba}$ implies the validity of the Newton's third law (action-reaction principle) as well. One can notice here the importance of the anti-symmetry flux: indeed, the anti-symmetry corresponds to the momentum conservation.
Similarly, the total angular momentum of the system with respect to the origin is given by

$$\underline{M} = \sum_{a=1}^{N} \underline{r}_a \times m_a \underline{u}_a \tag{2.60}$$

Again, time differentiation and the use of the equilibrium equation in the absence of external forces gives

$$\dot{\underline{M}} = \sum_{a=1}^{N} \underline{r}_a \times m_a \underline{a}_a = -\sum_{a=1}^{N} \underline{r}_a \times \underline{T}_a \tag{2.61}$$

Consequently, angular momentum will be preserved provided that the total angular momentum of the internal forces about the origin vanishes, that is

$$\sum_{a=1}^{N} \underline{r}_a \times \underline{T}_a = \underline{0} \tag{2.62}$$

where $\underline{T}_a$ is once again defined by equation (2.57).
Considering two particles a and b, the total moment of the two interacting forces about the origin can be found as

$$\underline{r}_a \times \underline{T}_{ab} + \underline{r}_b \times \underline{T}_{ba} = -\left(\underline{r}_b - \underline{r}_a\right) \times \underline{T}_{ab} = \underline{r}_{ab} \times \underline{T}_{ab} \tag{2.63}$$

since $\underline{T}_{ab} = -\underline{T}_{ba}$. This expression will vanish whenever the interaction force $\underline{T}_{ab}$ is co-linear with the vector $\underline{r}_{ab}$, which is the case of equations (2.51) and (2.52). Both these formulations hence conserve the angular momentum, provided a spherical kernel is used.
These conservation properties can also be derived in a variational framework [6], [84].

Viscous term in SPH formalism

- Viscous term stemmed from the momentum equation
  Speith et al [76] established a rigorous SPH modelling of the viscous term, based on the viscous part of the momentum equation. In Lagrangian formalism and with index notations, the i-component τ_i of the viscous part of the momentum equation is written:

$$\tau_i = \frac{1}{\rho}\frac{\partial T_{ij}}{\partial x_j} \tag{2.64}$$

  where the viscous tensor components T_{ij} are defined by

$$T_{ij} = \rho\nu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] \tag{2.65}$$

  for an incompressible flow. x_l ($1 \leq l \leq 2$ in 2D and $1 \leq l \leq 3$ in 3D) corresponds to the l-spatial direction and u_l the velocity component according to this direction. ν denotes the kinematic viscosity of the fluid. According to Einstein's conventions, a summation is performed over the repeated subscripts.

Considering equation (2.13), relation (2.64) relative to particle a becomes:

$$\tau_{i,a} = \frac{1}{\rho_a^2} \sum_b m_b \left(T_{ij,b} - T_{ij,a}\right) \frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}} \tag{2.66}$$

with

$$T_{ij,a} = \rho_a \nu_a \left(\partial_j u_{i,a} + \partial_i u_{j,a}\right) \tag{2.67}$$

$\partial_j u_{i,a}$, denoting the velocity gradient at particle a, can be written in SPH formalism according to:

$$\partial_j u_{i,a} = \left.\frac{\partial u_i}{\partial x_j}\right|_a = \frac{1}{\rho_a} \sum_b m_b \left(u_{i,b} - u_{i,a}\right) \frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}} \tag{2.68}$$

This modelling is very efficient to simulate flows characterised by important viscous effects [76]. However, it also causes huge calculation time: indeed, involving a double summation, the algorithm scales as N^2. An alternative consists in firstly calculating $T_{ij,a}$ with relations (2.67) and (2.68) and to store it. Secondly, the viscous term could be evaluated according to (2.66), which reduces by a factor of 2 the calculation time of viscous terms.

- Viscous term modelled by Morris

  For stability reasons (see part 2.8), the viscous term of equation (2.50) is commonly formulated as a mixture of a standard SPH first derivative with a finite-differences approximation for the velocity gradient. By considering the pressure gradient expression (2.16), the viscous term $\tau_{i,a}$ relative to particle a and defined by equation (2.64) can at first be represented by:

$$\tau_{i,a} = \frac{1}{\rho_a} \sum_b \frac{m_b}{\rho_b} \left(T_{ij,a} + T_{ij,b}\right) \frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}} \tag{2.69}$$

where the stress tensor components T_{ij} are defined by equation (2.65). Equation (2.69) is now re-written:

$$\tau_{i,a} = \frac{1}{\rho_a} \sum_b \frac{m_b}{\rho_b} \left[\mu_a \left(\left.\frac{\partial u_i}{\partial x_j}\right|_a + \left.\frac{\partial u_j}{\partial x_i}\right|_a\right) + \mu_b \left(\left.\frac{\partial u_i}{\partial x_j}\right|_b + \left.\frac{\partial u_j}{\partial x_i}\right|_b\right)\right] \frac{\partial w_h\left(r_{ab}\right)}{\partial x_j} \tag{2.70}$$

The velocity gradient is here computed with a finite-difference approximation by using a Jacobian transform [82]:

$$\left.\frac{\partial u_i}{\partial x_j}\right|_a = \frac{u_{i,a} - u_{i,b}}{r_{ab}} . \frac{x_{j,a} - x_{j,b}}{r_{ab}} \tag{2.71}$$

Equation (2.70) becomes:

$$\tau_{i,a} = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\left[\underbrace{\left(\frac{u_{i,a}-u_{i,b}}{r_{ab}}\right)\left(\frac{x_{j,a}-x_{j,b}}{r_{ab}}\right)}_{A1} + \underbrace{\left(\frac{u_{j,a}-u_{j,b}}{r_{ab}}\right)\left(\frac{x_{i,a}-x_{i,b}}{r_{ab}}\right)}_{A2}\right] \times \left(\mu_a+\mu_b\right)\frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}} \qquad (2.72)$$

This equation reveals at first that the i-component of the viscous force in SPH formalism is composed of two terms: the first one $A1$ is a tangential force and the second $A2$ is a normal force between particles.

For an incompressible flow, equation (2.71) gives:

$$\left.\frac{\partial u_j}{\partial x_j}\right|_a = \frac{u_{j,a}-u_{j,b}}{r_{ab}}.\frac{x_{j,a}-x_{j,b}}{r_{ab}} = 0 \qquad (2.73)$$

Moreover, the kernel derivative $\frac{\partial w_h(r_{ab})}{\partial x_j}$ can be written $\dot{w}_h\left(r_{ab}\right)\frac{x_{j,ab}}{r_{ab}}$, where $x_{j,ab} = x_{j,a} - x_{j,b}$. The term $A2$ times the kernel derivative then exhibits the product $\left(u_{j,a}-u_{j,b}\right)\left(x_{j,a}-x_{j,b}\right)$. Considering equation (2.73), the term $A2$ of equation (2.72) hence cancels and the i-component of the viscous force is written:

$$\tau_{i,a} = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\left[\left(\frac{u_{i,a}-u_{i,b}}{r_{ab}}\right)\left(\frac{x_{j,a}-x_{j,b}}{r_{ab}}\right)\right]\left(\mu_a+\mu_b\right)\frac{\partial w_h\left(r_{ab}\right)}{\partial x_j} \qquad (2.74)$$

Writing $u_{i,a} - u_{i,b} = u_{i,ab}$ and $x_{i,a} - x_{i,b} = x_{i,ab}$, equation (2.74) becomes:

$$\tau_{i,a} = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\frac{\mu_a+\mu_b}{r_{ab}^2}x_{j,ab}\frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}}u_{i,ab} \qquad (2.75)$$

In order to avoid a zero denominator when two particles are accidentally at the same position, a numerical parameter $\eta = 0.1h$ is added to r_{ab}^2:

$$\tau_{i,a} = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\frac{\mu_a+\mu_b}{r_{ab}^2+\eta^2}x_{j,ab}\frac{\partial w_h\left(r_{ab}\right)}{\partial x_{j,a}}u_{i,ab} \qquad (2.76)$$

Equation (2.76) is identical to the formula given by Morris (see equation (2.77)) for low Reynolds incompressible flows. In fact, this formulation seems to be valid for low and high Reynolds flows as well.

$$\nu\triangle\underline{u}_a = \sum_b \frac{m_b(\mu_a+\mu_b)\underline{r}_{ab}.\dot{w}_h(r_{ab})\underline{\epsilon}_{ab}}{\rho_a\rho_b\left(r_{ab}^2+\eta^2\right)}\underline{u}_{ab} \qquad (2.77)$$

where $\underline{u}_{ab} = \underline{u}_a - \underline{u}_b$. The expression used by Morris expresses the viscous force of the momentum equation through tangential forces between particles, which is physically meaningful. Moreover, it is shown in Appendix A that this modelling corresponds to $\nu\triangle\underline{u}$ by reconsidering the continuum mechanics formalism. This term exactly conserves linear momentum but not angular momentum and is much more computationaly efficient than the formulation (2.65). However, it seems that this formulation is not well suited for high Reynolds flows simulations. Indeed, in this case, particles clump together when they become too close to each other. To avoid this drawback, Monaghan established another formulation:

- Viscous force modelled by Monaghan
  In order to convert tangential forces to central forces which will repel particles which are getting too close to each other, Monaghan considered the following term

$$\nu\Delta\underline{u}_a \approx \sum_b \frac{16\nu}{\rho_a+\rho_b}\frac{\underline{u}_{ab}\cdot\underline{r}_{ab}}{r_{ab}^2+\eta^2}\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} \qquad (2.78)$$

  where ν can also be replaced by $\frac{1}{2}\left(\frac{\mu_a}{\rho_a}+\frac{\mu_b}{\rho_b}\right)$. Notice that in (2.77), the viscous force is colinear to the difference of velocity vectors $\underline{u}_{ab}$ whereas in (2.78) it is radial, aligned on $\underline{\epsilon}_{ab}$. It is proved in [93] that equation (2.78) corresponds to $\nu\triangle\underline{u}$ when reconsidering the continuum mechanics formalism. Due to the anti-symmetry of (2.78) with respect to subscripts a and b, this modelling conserves linear momentum. Conservation of angular momentum is also assumed in [84] and results from the spherical characteristic of the kernel. Moreover, in a solid rotation, the viscous term is rigorously zero. This term is often used when shock simulation or fast flows are considered. The idea to simulate shocks by using an artificial viscosity was at first used in Eulerian formulation [65]. However, it provides less intuitive formulation of viscous stresses which are not modelled through tangential forces.

- “Direct” viscous term
  Chaniotis [10] used direct differentiation of the kernel function to compute second-order derivatives. In order to remedy the sensitivity relative to the particle disorder (it will be show in part 2.8 that the particle disorder is very sensitive to the second derivative of the kernel), the particle locations are periodically reinitialised on a uniform grid. This process of remeshing has firstly been introduced in vortex methods. Nevertheless, for applications such as wave-flumes or dam breaking simulations, this “direct” viscous term is not suitable due to the remeshing procedure. Moreover, by remeshing the particles, the user partially loses the Lagrangian feature.

Considering the modelling of viscous forces established by Monaghan and Morris, the equation of motion in SPH formalism can respectively be written:

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{16\nu}{\rho_a + \rho_b} \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \underline{F}_a^e \tag{2.79}$$

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \sum_b \frac{m_b \left(\mu_a + \mu_b \right) \underline{r}_{ab} \cdot \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + \eta^2 \right)} \underline{u}_{ab} + \underline{F}_a^e \tag{2.80}$$

Specific smoothing length

When each particle is characterised by a specific smoothing length h, the momentum equation is obtained by considering the Lagrangian operator L of the system Ω [84]:

$$L = \int_\Omega \left[\frac{1}{2} \underline{\dot{r}}^2 - U \left(\rho \right) \right] \rho dr \tag{2.81}$$

where U denotes the internal energy. An evaluation of the previous equation with the Monte Carlo method gives:

$$L = \frac{M}{N} \sum_{j=1}^{N} \left[\frac{1}{2} \dot{r}_j^2 - U \left(\rho_j \right) \right] \tag{2.82}$$

In the case of an isentropic gas, Lagrange's equation relative to a particle a is:

$$\ddot{\underline{r}}_a = -\sum_{b=1}^{N} \left[\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right] \left(\frac{\partial w_h \left(r_{ab} \right)}{\partial \underline{r}_a} \right)_{h=cst} + \left(\frac{\partial}{\partial h} \sum_{b=1}^{N} -U \left(\rho_b \right) \right) \frac{\partial h}{\partial \underline{r}_a} \tag{2.83}$$

The term $\frac{\partial h}{\partial \underline{r}_a}$ varies in $\frac{1}{N^{2/d}}$ [23], where N corresponds to the particle number and d the problem dimension. Alimi [1] revealed that it should be used when irreversible processes such as energy dissipation due to shocks or dissipation by viscous effects are involved. However, according to estimations performed by Gingold, this term is probably negligible if a high number of particles is considered.

Variable smoothing length is the SPH analogue to adaptative gridding in mesh-based codes and allows greater resolution in regions where it is needed [43].

2.4.3 Pressure determination

State equation

The pressure of SPH particles can be obtained through a state equation as it was initially done in grid-based techniques to model steady incompressible flows [11]. In 1967, Batchelor

[58] established a state equation which accurately describes evolution of sound waves in water. Monaghan modified this equation so that it is able to deal with a lower speed of sound than the real one (this modification will be justified in part 2.5.1). He obtained the following state equation:

$$p(\rho) = B\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{2.84}$$

where

$$B = \frac{\rho_0 c_0^2}{\gamma} \qquad \textit{and} \qquad \gamma = 7 \tag{2.85}$$

ρ_0 and c_0 respectively denote the reference density and a numerical speed of sound. The choice of $\gamma = 7$ makes pressure strongly respond to density variation. Consequently, when particles are getting too close to each other, their pressure will highly increase and will repell these particles from each other through the pressure gradient term. This consideration, verified and presented in [37], is very important to avoid the tensile instability discussed in part 2.8. Therefore, particles remain in a suitable distribution and perturbations of the density field remain small, even at high Reynolds number [64]. However, small errors in density correspond to increasingly large errors in pressure. With this equation, the pressure field can be erroneous (see chapter 4). For low Reynolds number, more accurate pressure estimates are obtained if γ is taken to be unity, since errors in density and pressure remain proportional [64].
Through this equation, it comes that pressure automatically goes to zero when density equals the reference density. It consequently ensures the zero pressure condition relative to a free surface.

Strictly incompressible method

This part describes a strictly incompressible SPH formulation: here, the pressure is not an explicit thermodynamic variable as before but obtained by solving a pressure Poisson equation. The CFL condition (see part 2.5) is hence based on the fluid velocity field rather than the speed of sound. In [72], the authors use a predictor-corrector time scheme to compute the incompressible method:

- The first step (predictor step) is an explicit integration in time without enforcing the incompressibility. Only the viscous term and external forces are considered in the momentum equation. Intermediate particle velocity $\underline{u}^{n+\frac{1}{2}}$ and position $\underline{r}^{n+\frac{1}{2}}$ are firstly obtained according to

$$\underline{u}^{n+\frac{1}{2}} = \underline{u}^{n} + \delta\underline{u}^{n+\frac{1}{2}} \tag{2.86}$$

$$\underline{r}^{n+\frac{1}{2}} = \underline{r}^{n} + \underline{u}^{n+\frac{1}{2}}\delta t \tag{2.87}$$

where $\underline{u}^n$ (respectively $\underline{r}^n$) corresponds to the particle velocity (resp. position) at the time n, $\underline{u}^{n+\frac{1}{2}}$ (resp. $\underline{r}^{n+\frac{1}{2}}$) the intermediate particle velocity (resp. position) and $\delta\underline{u}^{n+\frac{1}{2}}$ the changed particle velocity during the predictor process defined by

$$\delta\underline{u}^{n+\frac{1}{2}} = \left(\underline{F}^{e} + \nu\Delta\underline{u}\right)\delta t \tag{2.88}$$

In order to simplify the notations, equation (2.88) has not been written in SPH formalism. Equations (2.77) or (2.78) can alternatively be used to write the viscous term. In this first step, incompressibility is not satisfied. Indeed, the fluid density $\rho^{n+\frac{1}{2}}$ based on the intermediate particle position $\underline{r}^{n+\frac{1}{2}}$ and the intermediate velocity $\underline{u}^{n+\frac{1}{2}}$ is different from the constant value ρ_0 [19].

- A second step (corrector step) is applied to adjust the particle density to the initial constant value ρ_0. In the correction, the pressure term is used to update the particle velocity obtained from the intermediate step according to

$$\underline{u}^{n+1} = \underline{u}^{n+\frac{1}{2}} + \delta\underline{u}^{n+\frac{1}{2}*} \tag{2.89}$$

where $\delta\underline{u}^{n+\frac{1}{2}*}$ corresponds to the changed particle velocity during the correction process defined by

$$\delta\underline{u}^{n+\frac{1}{2}*} = -\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+1}\delta t \tag{2.90}$$

with p^{n+1} denoting the particle pressure at the time $n+1$. The equation used to obtain the pressure for enforcing the incompressibility is the mass conservation which can be written in a discrete form as

$$\frac{1}{\rho_0}\frac{\rho_0 - \rho^{n+\frac{1}{2}}}{\delta t} + \nabla.\left(\delta\underline{u}^{n+\frac{1}{2}*}\right) = 0 \tag{2.91}$$

Combining equations (2.91) and (2.90), the pressure Poisson equation appears:

$$\nabla.\left(\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+1}\right) = \frac{\rho_0 - \rho^{n+\frac{1}{2}}}{\rho_0\delta t^2} \tag{2.92}$$

The source term of the Poisson equation is the variation of the particle density. Since the second derivative of the kernel is very sensitive to particle disorder (see part 2.8), the Laplacian of the Poisson equation is formulated as a hybrid of a standard SPH first derivative with a finite-differences approximation for the first derivative, as for the viscous term. The same approach has been used for the viscous term developed by

Monaghan (see part 2.4.2):

$$\nabla.\left(\frac{1}{\rho}\nabla p\right)_a = \sum_b m_b \frac{8}{(\rho_a+\rho_b)^2} \frac{p_{ab}\underline{r}_{ab}.\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab}}{r_{ab}^2+\eta^2} \tag{2.93}$$

where $p_{ab} = p_a - p_b$. This form has the advantage to be anti-symmetric with respect to a and b subscripts. Equation (2.93) is then discretised into linear equations which are efficiently solved by using a preconditioned conjugate gradient method, like in finite element methods.

Finally, the particle positions are centered in time according to

$$\underline{r}^{n+1} = \underline{r}^{n} + \frac{\left(\underline{u}^{n} + \underline{u}^{n+1}\right)}{2}\delta t \tag{2.94}$$

where $\underline{r}^n$ and $\underline{r}^{n+1}$ respectively correspond to the positions of particle at the time n and $n+1$. Boundary and free surface conditions relative to the strictly incompressible method are presented in [72].

A strictly incompressible method increases the stability of calculations and the time step can be at least ten times smaller. However, boundary conditions require more investigation and more importantly the linear equation resolution can be computationnaly heavy.

2.4.4 Moving SPH particles

Once the particle velocity has been determined, several methods enable to compute the particle position.

Simple temporal integration

Considering the definition of the velocity,

$$\frac{d\underline{r}_a}{dt} = \underline{u}_a \tag{2.95}$$

the new position of particle a can be obtained by a simple temporal integration of its velocity (numerical methods for temporal integration are presented in part 2.5).

XSPH method

For free surface flows, it is often efficient to determine the particle position according to the XSPH method ([56], [57], [58]):

$$\left.\frac{d\underline{r}_a}{dt}\right|_{XSPH} = \underline{u}_a + \underbrace{\epsilon \sum_b \frac{m_b\left(\underline{u}_b - \underline{u}_a\right)}{\bar{\rho}_{ab}} w_h\left(r_{ab}\right)}_{corrective\quad term} \tag{2.96}$$

where

$$\bar{\rho_{ab}} = \frac{\rho_a + \rho_b}{2} \qquad \mathit{and} \qquad \epsilon \in [0;1[\tag{2.97}$$

Contrary to the first method, the position of particle a is not only obtained through its proper velocity $\underline{u}_a$. Indeed, due to the corrective term, it is modified with an averaged velocity characteristic of the neighbourhood of particle a. The new velocity differs from the former one by a term of order $O\left(h^2\right)$, which is consistent with the errors of the other SPH equations. In order to investigate the effect of the corrective term, Gray [24] established a SPH dispersion relation by stacking a small perturbation to equations (2.95) and (2.96) according to

$$\frac{d\underline{r}_a}{dt} = \underline{V} e^{i(\underline{k}.\bar{\underline{r}}_a - \omega t)} \tag{2.98}$$

$$\left.\frac{d\underline{r}_a}{dt}\right|_{XSPH} = \bar{\underline{V}} e^{i(\underline{k}.\bar{\underline{r}}_a - \omega t)} \tag{2.99}$$

where $\bar{\underline{r}}_a$ corresponds to the initial particle position. Considering equation (2.96) with $\epsilon = 1/2$, the relation between $\underline{V}$ and $\bar{\underline{V}}$ can be simplified for long wavelengths according to

$$\bar{\underline{V}} = \frac{1}{2}\underline{V}\left(1 + \tilde{W}\right) \tag{2.100}$$

where $\tilde{W}$ denotes the Fourier transform of w_h. This expression shows that $\bar{\underline{V}}$ introduces additional dispersion coming from the Fourier transform of the kernel.

XSPH method has proved its capability ([56], [57], [64]) when simulating nearly incompressible flows: by increasing dispersion, it avoids situations where particles clump together due to numerical instabilities. Thus, particles stay more ordered. Moreover, in high speed flow, it prevents the penetration of one fluid by another [58]. Indeed, if $\underline{u}_b = -\underline{u}_a$, particles a and b can be moved at the same position without the corrective term.

With this method, each particle is characterised by two velocities: the first one results from the momentum equation and the second corresponds to the r.h.s. of equation (2.96). Therefeore, to ensure the consistency of Navier-Stokes equations, the velocity used in the continuity equation (2.45) must be the corrected velocity.

2.5 Temporal integration

2.5.1 Time step

The time step must take into account:

- a CFL condition

- the viscous diffusion term
- the magnitude of internal and external forces affected to each particle

CFL condition

The CFL condition imposes that the time step δt_{CFL} is less or equal to the convection time on the length h relative to the spatial discretisation. In SPH formalism and in hydraulics applications ($M << 1$), this condition is expressed by

$$\delta t_{CFL} = 0.4 \frac{h}{c_0} \tag{2.101}$$

where the coefficient 0.4 was determined through numerical tests [56]. c_0 corresponds to a numerical speed of sound characteristic of the flow. The aim of this condition is to avoid artificial pressure wave propagation due to the state equation.

Condition in terms of internal and external forces

This condition ensures that particles do not get too close to each other during the integration of their movement [64]:

$$\delta t_{forces} = 0.25 \times \min_a \sqrt{\left(\frac{h}{|\underline{f}_a|} \right)} \tag{2.102}$$

where $\underline{f}_a$ denotes the internal and external forces associated to particle a (i.e. the magnitude of the r.h.s. of the momentum equation (2.50)).

Condition in terms of the viscous term

The viscous criterion must be taken into account to make the time step inferior to the viscous phenomenon time scale. In a finite volume or finite element code [83], this time scale is based on a characteristic cell size, which corresponds here to the smoothing length. The viscous criterion could so be defined by

$$\delta t_{visc} = 0.125 \frac{h^2}{\nu} \tag{2.103}$$

At sufficiently high resolution (small h) or large viscosity, relation (2.103) is typically the dominant time constraint [64]. This condition is generally unnecessary, except for low Reynolds flows [64] characterised by $Re < 3.2\ M$. Re is a Reynolds number based on the smoothing length and a typical velocity of the flow and M denotes the Mach number.

Final time step

The final time step, denoted δt, is obtained according to:

$$\delta t = \min\left(\delta t_{forces}, \delta t_{CFL}, \delta t_{visc}\right) \tag{2.104}$$

Choice of the speed of sound

It is important to notice that the time step dictated by the CFL condition directly depends upon the speed of sound c_0. Consequently, if the user chooses for c_0 its real value (for water, c_0 = 1 400 $m.s^{-1}$ in normal temperature and pressure conditions), the time step would be very low. When simulating a nearly incompressible flow, it is thus practical to choose an artificial lower value for the speed of sound [56], [64], in order to ensure an efficient solution for a given problem. In this case, the flow is no more incompressible but nearly incompressible, provided that the chosen value of c_0 is suitable. A dimensional analysis of the momentum equation and the state equation give:

$$c_0^2 \propto \frac{V_0^2}{\delta} \tag{2.105}$$

where δ denotes the relative density variation $\delta\rho/\rho$ and V_0 a characteristic velocity of the flow. Consequently, the user can impose c_0 in order to deal with the required value of δ for a nearly incompressible flow: generally, it is acceptable to consider $\delta \approx$ 1%; c_0 must be then at least ten times superior to the maximal velocity of the flow, which corresponds to a Mach number less than 0.1.

2.5.2 Temporal integration schemes

The momentum, continuity and position equations can be written in the following generalised forms:

$$\begin{cases} \frac{d\underline{u}_a}{dt} = \underline{F}_a \\ \frac{d\rho_a}{dt} = \underline{G}_a \\ \frac{d\underline{r}_a}{dt} = \underline{H}_a \end{cases} \tag{2.106}$$

where quantities $\underline{G}_a$ and $\underline{H}_a$ respectively correspond to the r.h.s. of equations (2.45) and (2.95) (or (2.96)). Concerning the quantity $\underline{F}_a$, it is obtained by considering the r.h.s. of equations (2.79) or (2.80). Several methods enable to integrate the Lagrangian derivative present in the coupled system (2.106):

Explicit scheme

Integration with the explicit scheme gives:

$$\left\{\begin{aligned} \underline{u}_a^{n+1} &= \underline{u}_a^n + \delta t \underline{F}_a^n \\ \rho_a^{n+1} &= \rho_a^n + \delta t \underline{G}_a^n \\ \underline{r}_a^{n+1} &= \underline{r}_a^n + \delta t \underline{H}_a^{n+1} \end{aligned}\right. \tag{2.107}$$

where $\underline{A}^n$ (resp. $\underline{A}^{n+1}$) corresponds to the value of $\underline{A}$ quantity at the time n (resp. $n+1$). This scheme is obviously of order one in δt.

Predictor corrector scheme

Applied to the momentum equation (it is the same for continuity and position equations), this numerical scheme can be written [58]:

- Predictor step

$$\underline{u}_a^{n+\frac{1}{2}} = \underline{u}_a^n + \frac{1}{2}\delta t \underline{F}_a^n \tag{2.108}$$

- Corrector step

$$\underline{u}_a^{n+\frac{1}{2}*} = \underline{u}_a^n + \frac{1}{2}\delta t \underline{F}_a^{n+\frac{1}{2}} \tag{2.109}$$

- Final step

$$\underline{u}_a^{n+1} = 2\underline{u}_a^{n+\frac{1}{2}*} - \underline{u}_a^n \tag{2.110}$$

With this method, integration errors are of order δt^3 but calculation time is doubled.

Implicit scheme

Since a given SPH particle can have approximately 40 neighbours contributing to the evaluation of any property, any implicit scheme will have matrices which would be theoretically banded with approximately 40 elements wide. Starting with the natural numbering of particles, the matrix would be extremely sparse and a numerical scheme based on solving these matrix equations by a direct method would be impractical. As the particles move at each time step, the connectivity changes and each iteration would require a re-ordering of the matrix before attempting Gauss-Seidel resolution for instance. To avoid this drawback, Monaghan [60] treats each pair of interaction separately: the velocities of the two particles involved is implicitly updated according to their pair interaction. This process is repeated for each pair interaction. The results obtained in [60] are satisfactory in gas dynamic problems but further developments are required in order to take into account viscous effects.

Integration scheme with individual time steps

In most SPH codes, all the particles are simultaneously advanced at each time step. The particle needing the highest time resolution determines the timestep of all the others. To make a code more efficient in handling problems with multiple time scales, the computational effort must be centered on those particles that require it, avoiding useless computations for the remaining particles. In other words, it is necessary to allow different timesteps for each particle. This has been done in the SPH code TREEASPH. The procedure is described in details in [1]. This new technique considerably increases the efficiency of a code.
Elaborate numerical schemes are nowadays implemented in SPH codes: for instance, Flebbe [21] uses a Runge-Kutta-Fehlberg method which is a high precision combined fifth- and fourth-order Runge-Kutta with an adaptive step size.

2.6 Initial particle distribution

2.6.1 Complete stochastic distribution

When simulating a viscous gas ring around a central mass, Speith [76] reveals the importance of the initial particle distribution: three initial configurations tested are presented on figure 2.12:

1. Particles are initially placed on concentric spheres around the origin, as shown on picture $a1$ of figure 2.12. During the simulation, this uniform distribution is altered even if it still remains quite coherent (see picture $a2$ of figure 2.12) .

2. Particles are initially disposed in a stochastic way according to a probability density which is close to an initial theoretical density (picture $b1$ of figure 2.12). The system does not present any symmetry point and after some time, instabilities in the form of spirals perturb the calculation, as can be seen from picture $b2$ of figure 2.12.

3. The third configuration ($c1$ on figure 2.12) is identical to the previous one but presents a central symmetry point. In this case, perturbations which induce spirals structures are considerably reduced (see picture $c2$ of figure 2.12).

Since a complete stochastic distribution of the particles induces large fluctuations in the initial density field [97], the third initial configuration was adopted by the author in order to simulate the dynamic system evolution. In a general way, fluid dynamic must be simulated from an initial organised and physical configuration [58]. This is consistent with the fact that high density zones should concentrate more particles. If it is not the case, fluctuations associated

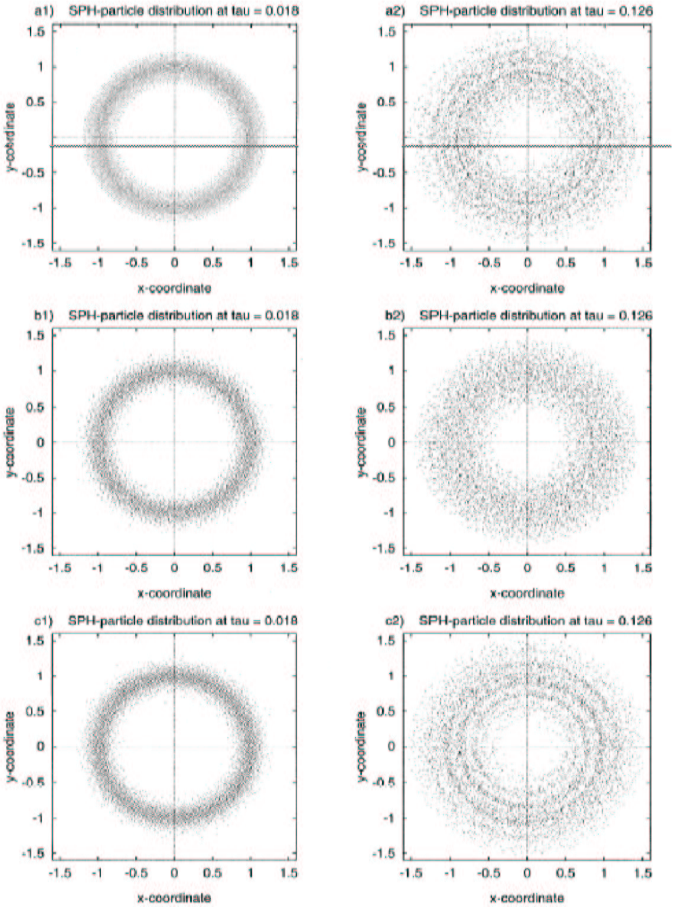

Figure 2.12: $a1$, $b1$, $c1$: initial configurations ; $a2$, $b2$, $c2$: evolution at $t = 0.126$ s [76].

to particle position will be involved in initial conditions and the calculation will be perturbed. Moreover, these perturbations will be larger if non-linear phenomena are present.

2.6.2 Evolution mechanism towards a steady state

The pressure gradient of the momentum equation corresponds to a repulsive force between two particles. This force can thus be linked to a potential energy [97] which is minimum when particles are equidistant (this can also be shown from the Appendix B). Therefore, the initial distribution of particles should evolve towards this minimum energy state and this tends to iron out any random density fluctuations.

- Due to an artificial viscosity [97], any initial particle distribution can evolve towards this minimum energy state and previous density fluctuations are considerably reduced.

- Even if the user manually adjusts the particle density in order to dispose of an initial steady state, the system can contain residual forces which are not balanced [59]. To obtain a real steady state, a damping term Γ ([23], [56]) could be introduced in the momentum equation in order to dissipate the residual kinetic energy:

  $$\frac{d\underline{u}}{dt} = \underline{F} - \Gamma\underline{u} \tag{2.111}$$

  Without external volumetric force such as gravity or magnetic force, particles can adopt a crystallized structure after a relatively long time.

- In order to settle the particles when an inital random distribution is considered, Whitworth [97] revises their position according to the following iterative routine

  $$\underline{r}_a^{n+1} = \underline{r}_a^n + \epsilon h \sum_b \left[\left(\frac{\rho_b - \rho_{true}}{\rho_{true}} \right) w_h \left(r_{ab} \right) \frac{(\underline{r}_a - \underline{r}_b)}{r_{ab}} \right] \tag{2.112}$$

  where ρ_{true} corresponds to the mean density of the system equals to Nm/H^3. N is the total number of particles, m the mass of a particle and H^3 the volume of the 3D computation domain. With this relaxation formula, he showed that 100 iterations were enough, with $\epsilon = 1$. Figure 2.13 reveals that the density ditribution after a settling phase (right picture) is much more accurate than the one without it (left picture). In this figure, ρ_{true} corresponds to the theoretical value and $f(x)dx$ is the fraction of particles in the range $(x, x + dx)$.

2.7 Wall modelling

Microscopically, walls are constituted by atoms which exert repulsive forces on fluid. Inspired by this physical principle, walls can be modelled in SPH theory by solid particles which exert

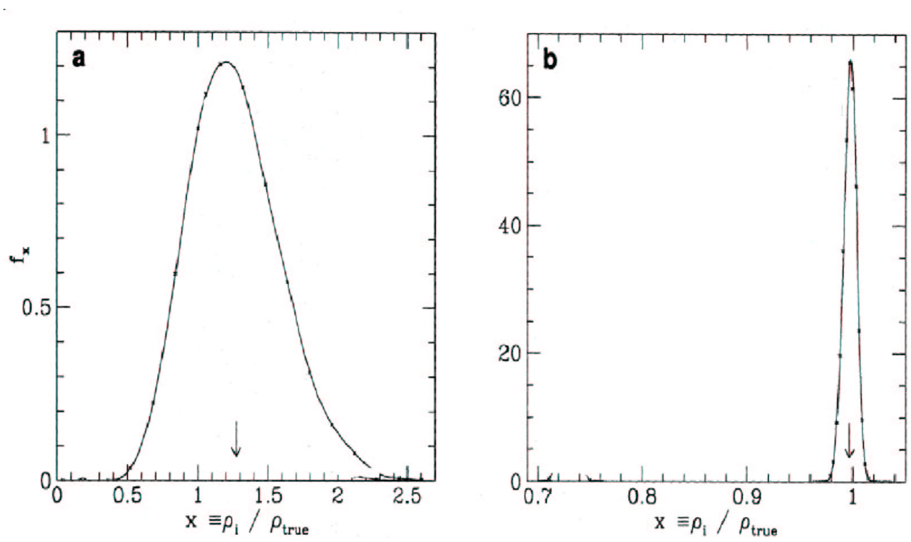

Figure 2.13: Normalised density distributions: ***a***: for an initial random distribution. ***b***: after settling [97].

repulsive forces on fluid particles ([56], [57]), ensuring thus wall impermeability. These solid particles, called wall particles, are not involved in the pressure gradient term of the momentum equation. However, if zero velocity at the wall is required, they contribute to the viscous term in order to mimic no-slip conditions. The wall particles are immobile or can define a mobile wall. In this last case, their velocity is equal to the wall velocity: for instance, Monaghan [58] considered a moving lock gate modelled by a set of moving wall particles.

Three types of forces among several are presented in this part: a purely repulsive force, a similar Lennard-Jones force and a normal force. These forces are added in the r.h.s. of the momentum equation (2.79) (or (2.80)), as external forces.

2.7.1 Repulsive force

Purely repulsive force

A purely repulsive central force $\underline{f}_1$ exerted by a wall particle p on a fluid particle a can be written:

$$\underline{f}_1\left(r_{ap}\right) = \begin{cases} D\left[\left(\frac{r_0}{r_{ap}}\right)^{p_1} - \left(\frac{r_0}{r_{ap}}\right)^{p_2}\right]\frac{\underline{r}_{ap}}{r_{ap}^2} & \text{if} \quad r_{ap} \leq r_0 \\ 0 & \text{if} \quad r_{ap} > r_0 \end{cases} \tag{2.113}$$

where r_0 denotes the wall influence radius and D corresponds to a numerical parameter. Microscopically, the radius of repulsive forces is approximately equal to the atomic radius. In SPH, it corresponds to a characteristic particle length scale, i.e. δr. Another definition of this type of force is presented by Monaghan in [58]. The force action is illustrated on figure 2.14. A drawback of this type of forces is the involvement of several adjustable constants which are

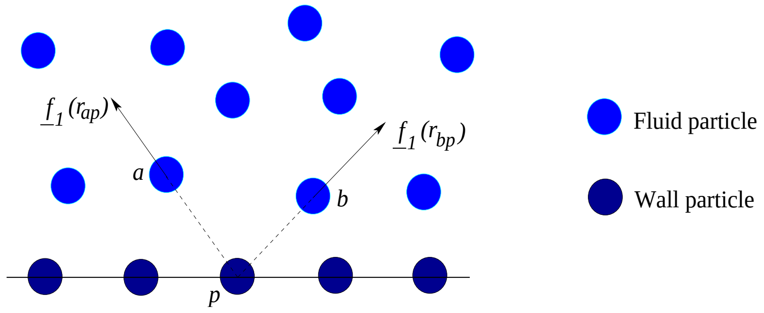

Figure 2.14: Action of a purely central repulsive force.

empirically determined:

- In most cases, sets $p_1 = 4$, $p_2 = 2$ or $p_1 = 12$, $p_2 = 6$ are suitable [56]. The most important feature is $p_1 > p_2$.

- Concerning the D parameter, it depends on the physical problem. For a flow driven by the gravity, if H defines a characteristic vertical length scale of the problem, a value of D around gH is suitable [57].

Lennard-Jones force

Another central wall force, inspired by the Lennard-Jones interatomic force, can be developed. Contrary to the previous one, this force presents an attractive core:

$$\underline{f}_2\left(r_{ap}\right) = \begin{cases} \epsilon\left[\left(\frac{l}{r_{ap}}\right)^m - \left(\frac{l}{r_{ap}}\right)^n\right] & \text{if} \quad 0 < r_{ap} < r_c \\ A\left(R - r_{ap}\right)^2 + D\left(R - r_{ap}\right) & \text{if} \quad r_c < r_{ap} < R \\ 0 & \text{if} \quad r_{ap} > R \end{cases} \tag{2.114}$$

where r_c represents the distance which separates the attractive part from the repulsive part of the force. Values of the different parameters of equation (2.114) are defined by Monaghan in [59]. One problem associated with central forces is that they produce unwanted perturbations in the flow due to their discreteness. Indeed, it is known that replacing a straight boundary by an undulating one results in more wall perturbations, even if the amplitude of the oscillations is small: when investigating a gravity current, Monaghan [59] noticed that the gravity current thins out to a thickness of $5h$ to $6h$ due to the discreteness in the boundary force.

Normal repulsive force

In order to work with smoother wall forces, Monaghan [59] established a force characterised by a normal wall direction. This third type of force, based on the normal and tangential distances of the fluid particle to wall particles can be written:

$$\underline{f}_3\left(r_{ap}\right) = f_1\left(\underline{n}.\underline{r}\right) P\left(\underline{t}.\underline{r}\right) \underline{n} \tag{2.115}$$

$\underline{n}$ and $\underline{t}$ respectively denote the normal and tangential vectors to the wall, as shown on figure 2.15. The P function involves the tangential distance between the fluid particle a and the wall particle w. It hence gives to the wall a continuous aspect when a fluid particle gets close to several wall particles. If δr represents the initial distance between two wall particles, P can be expressed as:

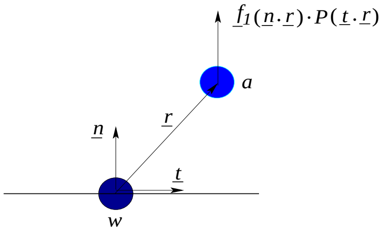
$\underline{f}_1(\underline{n}\cdot\underline{r})\cdot P(\underline{t}\cdot\underline{r})$
a
$\underline{r}$
$\underline{n}$
$\underline{t}$
w

Figure 2.15: Action of a normal repulsive force.

$$P\left(q\right) = \begin{cases} 1 - \frac{|q|}{\delta r} & \text{if} \quad 0 < |q| < \delta r \\ 0 & \text{if} \quad |q| > \delta r \end{cases} \tag{2.116}$$

where $q = \underline{t}.\underline{r}$. This wall force expression is generalised for curved walls by Monaghan [59] and gives good results.

Comparison of wall forces

- According to Monaghan's investigations ([56], [59]), velocity fields obtained with the radial wall forces $\underline{f}_1$ and $\underline{f}_2$ conform to experimental results even if these two forces induce particle disorganisation close to the wall. This is due to numerical instabilities associated to wall forces (see part 2.8.3).

- Particle disorganisation close to the wall due to $\underline{f}_1$ and $\underline{f}_2$ is considerably reduced if the spacing between wall particles decreases. Indeed, closer wall particles tend towards a smooth wall. Monaghan showed that results are much more satisfactory if wall particles are four times more packed than fluid particles in the initial state ([63], [59]).

- With the force $\underline{f}_3$, velocity field is smoother than the one obtained with forces $\underline{f}_1$ and $\underline{f}_2$ when wall particles are four times closer. Moreover, it is much cheaper to store normal and tangential distances than taking into account four times more wall particles.

As a conclusion, normal forces seem to be the most efficient ones.

2.7.2 Mirror particles

For the high Reynolds number free surface flows considered by Monaghan in [59] and [57], free slip boundary conditions are used. However, to realistically model flows at lower Reynolds number, no slip boundary conditions are required. Since wall forces are not accurate enough to obtain a satisfactory friction coefficient, a different approach with mirror particles has been developed.

No slip boundary conditions

Morris describes in [64] a simple and quite accurate wall modelling method based on mirror particles. In this case, there is no direct discretisation of the wall. Mirror particles are defined as those initially lying inside obstacles within the flow field and beyond solid walls. As represented on figure 2.16, for each fluid particle a, the normal distance d_a to the boundary is firstly calculated. This normal distance is then used to define a tangent plane (a line in 2D) from which the normal distance d_B to each mirror particle B is computed. The velocity of particle a is extrapolated across the tangent plane, assuming zero velocity on the plane itself, thus giving each mirror particle the velocity $V_B = -(d_B/d_a)V_a$. In practice, the discrete arrangement of mirror particles may allow a fluid particle to closely approach the curve describing the boundary. In such circumstances, the magnitude of V_B must be restricted. Accordingly, the following formula is used to calculate the mirror particle velocity:

$$V_B = V_a\,(1-\beta) \tag{2.117}$$

with $\beta = min\left(\beta_{max}, 1 + \frac{d_B}{d_a}\right)$. Numerical simulations have shown that good results are obtained if β_{max} is approximately 1.5. The artificial velocity V_B is used to calculate viscous forces, but it is not used to displace mirror particles.

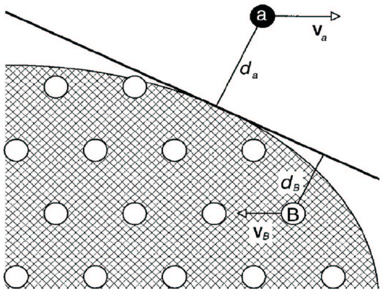

Figure 2.16: Mirror particle velocity for a curved wall [64].

Impermeability ensured by symmetry consideration

A simpler formulation than the one used by Morris has been developed during this work. The wall is modelled with wall particles and extra layers of mirror particles are added under the wall: the density of wall and mirror particles evolves thanks to the contribution of fluid particles through the continuity equation (2.45). Indeed, if the fluid particle a is linked to the wall or mirror particle b, the b-contribution to the evolution of a-density is equal to:

$$m_b \underline{u}_{ab} \dot{w}_h(r_{ab}) \underline{\epsilon}_{ab} \tag{2.118}$$

As the particle b is also linked to the fluid particle a, the contribution of the particle a to the evolution of b-density is also given by equation (2.118), since the continuity equation (2.45) is symmetric with respect to a and b subscripts. The pressure of wall and mirror particles are then computed with the state equation and these particles are involved in the pressure gradient relative to fluid particles. To reach the same level of accuracy in computing the forces on fluid particles located in the vicinity of the wall, three or four layers of mirror particles are considered. These new wall conditions also enable a perfect impermeability of the wall in rapid dynamic phenomena such as dam breaking. Contrary to the repulsive forces previously described by equations (2.113) to (2.116), the present formulation does not introduce any ad hoc coefficient. In addition, it allows the treatment of complicated walls very easily. However, despite their ability to model walls precisely, the mirror particles slightly increase the CPU time.

2.8 SPH numerical instabilities

SPH numerical instabilities induce a typical and unphysical particle disorganisation: particles clump in a random way. This evolution is obviously inconsistent with physical laws at a macroscopic scale: in a real material, repulsive forces between atoms would prevent the particle clumping. The consequences of these instabilities are dramatic in “The Tennis Ball Problem” where SPH computations predicted weird and unphysical fragmentation of tennis balls in impacts encountered in typical tennis tournaments [32]. In industrial solid mechanics simulations, these instabilities play a prominent part since they could be mistaken for a physical fragmentation of the solid. Numerical instabilities also induce an exponential increase of the particle velocity [77] and a 1% perturbation on the density evaluation: in a nearly incompressible flow, it hence induces large pressure fluctuations due to the state equation. Figure 2.17 shows a typical clumping instability occured in a 2D free surface channel during a settling process. This process will be investigated in detail in chapter 4.

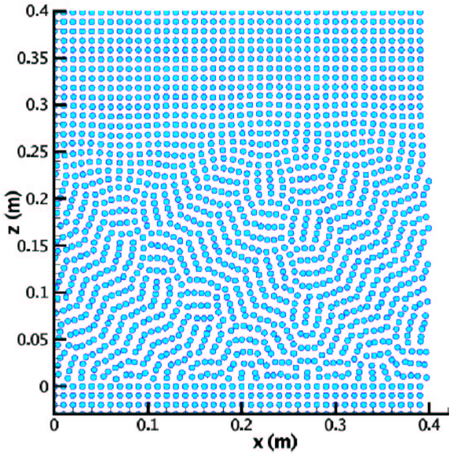

Figure 2.17: Tensile instability in a 2D free surface channel.

2.8.1 The tensile instability

If an elastic solid is stretched, SPH particles attract each other (through the pressure gradient term) in order to maintain the system cohesion. Even if this representation is consistent with the atom behaviour, this mutual particle attraction can induce numerical instabilities ([64], [77]). In the same way, when a fluid is described by a state equation susceptible to cause negative pressure which attract particles, numerical instabilities may occur ([62], [64]). These instabilities are known as the *tensile instability.* A one-dimensional Von-Neumann analysis has been achieved by Swegle in [77] in order to investigate SPH equation stability. After stacking a small perturbation to the Navier-Stokes equations, a propagation equation relative to the perturbation is obtained. The following sufficient instability condition has been deduced from the propagation equation:

$$\ddot{w}_h . T > 0 \tag{2.119}$$

where $\ddot{w}_h$ denotes the kernel second derivative and T the particle stress. By convention, $T < 0$ corresponds to a compression state and $T > 0$ a tensile state. This condition is represented on figure 2.18, where W' correspond to the first derivative of the kernel. Within a fluid, where T is normally negative (except numerical errors), one can notice that a kernel with a negative second derivative constitutes a sufficient instability condition. The picture on the right of figure 2.8 exhibits the negative portion of the second derivative of the third, fourth and fifth order spline kernels. Generally (whatever the dimensions of the problem), a more universal instability criterion can be deduced by considering the system potential energy. The corresponding proof due to Violeau [86] (simpler than the one achieved by Swegle and presented in Appendix A) reveals that criterion (2.119) is also a necessary instability condition. Both derivations of relation (2.119) are independent of the viscous term modelling and the

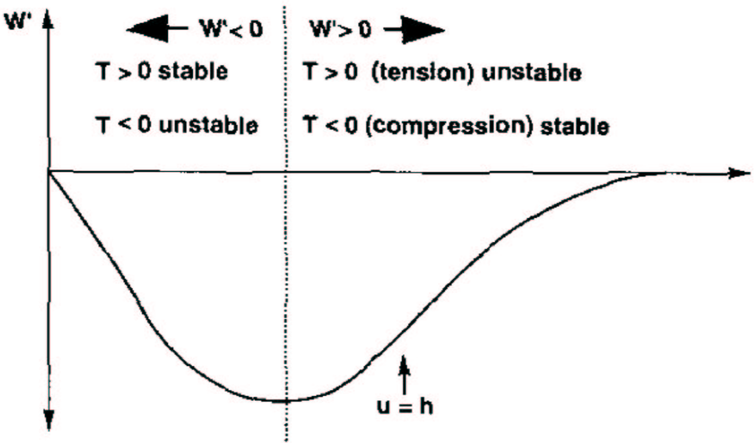

Figure 2.18: Sufficient instability condition [77].

temporal integration scheme [77].

2.8.2 Removing the tensile instability

There have been several attempts to remove the tensile instability: Gray listed some of them in [24] but revealed that there is no universal solution for all problems. Three solutions are suggested in this part.

Kernel improvement

Numerical instabilities due to a tensile state can be reduced by using an higher order kernel. Indeed, the picture on the right of figure 2.8 shows that the negative second derivative part of spline kernels is smaller when the order of the kernels increases. An illustration of this phenomenon will be given in chapter 4. Nevertheless, calculation time is obviously increased when higher order kernels are considered.

Artificial pressure

Since the tensile instability induces particle clumping, it can be qualified as a short wavelength instability. Hence, Monaghan [62] revealed that it can be removed if a short-range artificial force between particles is introduced in the momentum equation. The pressure gradient term present in the momentum equation (2.79) is thus modified according to:

$$\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \qquad \rightarrow \qquad \frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} + R f_{ab}^n \tag{2.120}$$

It is natural to define the function f_{ab} with respect to the kernel:

$$f_{ab} = \frac{w_h\left(r_{ab}\right)}{w_h\left(\delta r\right)} \tag{2.121}$$

In the added term, n must be positive and R is determined from a dispersion relation [62], [63]. δr corresponds to the initial distance between particles. Considering $n = 4$, $h = 1,3.\delta r$ and a third order spline kernel, Monaghan established that the repulsive force between two particles is 23 times more important when r_{ab} decreases from δr to 0. Conversely, the term $w_h\left(r_{ab}\right)/w_h\left(\delta r\right)$ briefly decreases if $h \leq r_{ab} \leq 2h$: in this zone, the cubic spline kernel decreases according to $\left(2 - r_{ab}/h\right)^3$ whereas the repulsive term decreases according to $\left(2 - r_{ab}/h\right)^{3n}$. The parameters of the force minimise the error in the long-wavelength modes and only nearest neighbours are thus influenced by the artificial pressure. The resulting algorithm is simple, accurate and effective.

The dynamic evolution of a bi-dimensional fluid disc was investigated by Monaghan in [62]. After 1 000 time steps, the particle clumping due to numerical instabilities occurs (see the left picture of figure 2.19). However, by introducing an artificial pressure such as described at equation (2.120), particles still remain organised (see the right picture of figure 2.19). In solid mechanics, simulation of two rubber cylinders collision is also very sensitive to numerical instabilities [62]. However, artificial pressure enables to successfully perform this simulation, as shown on figure 2.20.

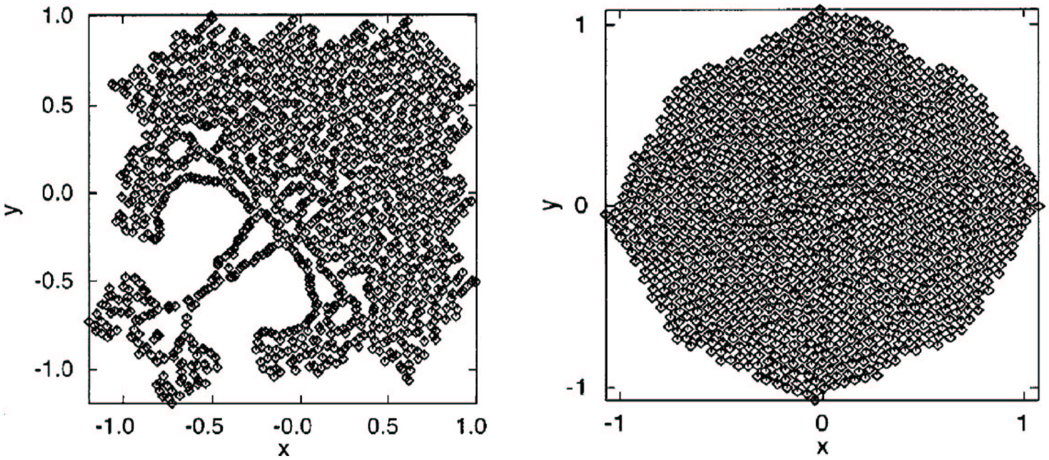

Figure 2.19: Influence of the artifical pressure on the tensile instability (left: without, right: with) [62].

Stabilisation with Shape-Shifting

SPH can also be stabilised by shifting the shape of w_h to change the sign of $\ddot{w}_h$ [32]. The drawback of this approach is that it can damp more short wavelength structure than desired.

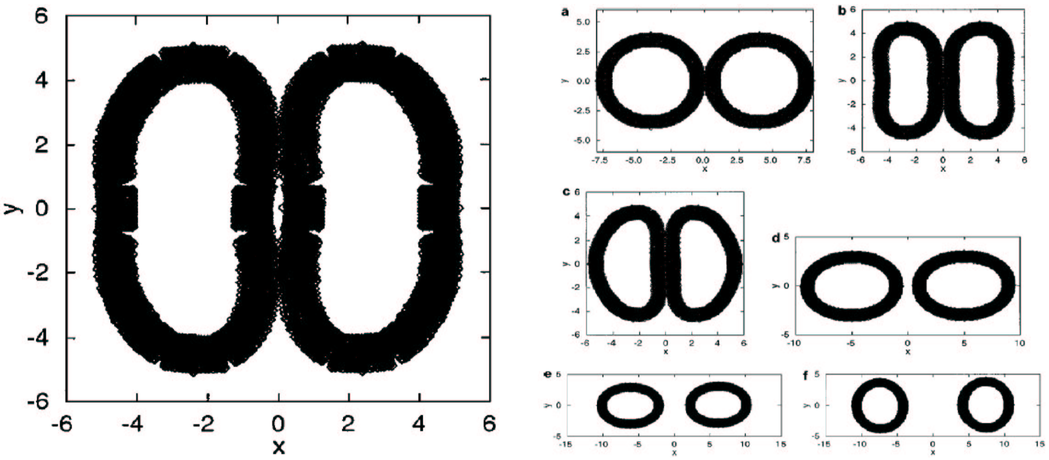

Figure 2.20: Influence of the artifical pressure on the tensile instability (left: without, right: with) [62].

Moreover, this method has been applied in one dimension and is quite cumbersome. For more details, the reader can refer to [32].

2.8.3 Numerical instabilities due to repulsive wall forces

As mentioned in part 2.7, repulsive central wall forces can induce large perturbations on the particle movement: effects of these perturbations are similar to those due to the tensile instability. These perturbations are linked to large variations of the normal component of the repulsive wall force applied to a fluid particle during its movement [59]. Evolution of the normal and horizontal components, respectively F_y and F_x, of a wall force applied to a fluid particle evolving at a fixed distance from the wall is represented on figure 2.21 [59]. This evolution is not surprising: the horizontal contributions of the wall force due to two wall particles seems to cancel or to be very small whereas the vertical contributions are added, as one can see from figure 2.22. A finer wall discretisation efficiently reduces numerical instabilities. Indeed, by considering four times more wall particles and consequently considering a wall force four times weaker, variations of the vertical component force are no longer visible on figure 2.21 [59]. In the case presented by Monaghan, particles remain ordered with this new discretisation since the closely packed boundary continues to give a smoother boundary velocity. The drawback of this close packing is that it rapidly becomes computationally inefficient since the number of particles increased, as mentioned before [59].

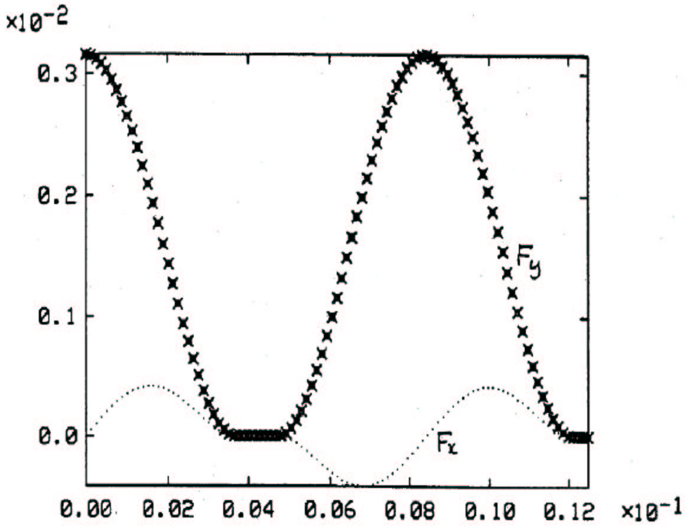

Figure 2.21: Evolution of the vertical (F_y) and horizontal (F_x) component (versus the wall coordinate) of a repulsive wall force for a particle evolving at a fixed distance from the wall [59].

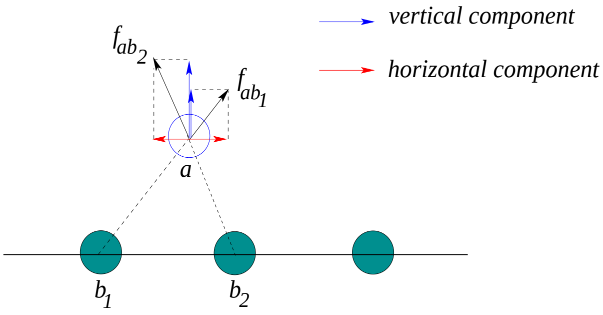

Figure 2.22: Vertical and horizontal component of a repulsive force.

2.9 SPH and Eulerian methods

2.9.1 Similarities

- It can be shown that fluid mechanics equations written in SPH formalism are very close to those discretised by finite-differences methods ([56], [57], [58]): on figure 2.23, x corresponds to the spatial direction and t the time. An explicit centred discretisation of the continuity equation (2.44) at the point (i, n) of the mesh gives

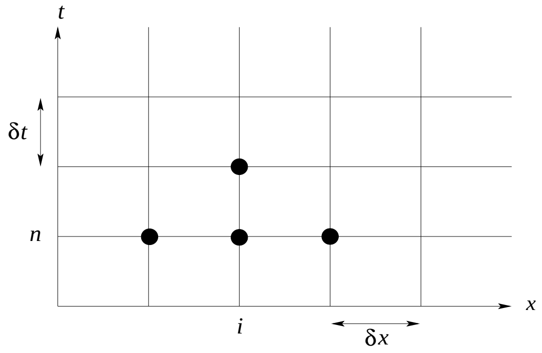

Figure 2.23: $1D$ schematic mesh used in Eulerian code.

$$\frac{\rho_i^{n+1} - \rho_i^n}{\delta t} + \frac{(\rho u)_{i+1}^n - (\rho u)_i^n}{\delta x} + O\left(\delta t, \delta x\right) = 0 \tag{2.122}$$

While an explicit SPH discretisation of the temporal term of the continuity equation (2.45) gives

$$\frac{\rho_a^{n+1} - \rho_a^n}{\delta t} + \rho_a \sum_b m_b \left(\underline{u}_b^n - \underline{u}_a^n\right) \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} + O\left(\delta t, \delta x^2\right) = 0 \tag{2.123}$$

It can also be noticed that the smoothing length h which describes the spatial discretisation in SPH formalism is analogous to the spatial step δx in finite differences.

- The use of different kernels in SPH is analogous to the use of different numerical schemes in Eulerian codes, as explained in part 2.3.3. Kernel functions are also analoguous to the shape functions used in finite element method. However, SPH has a numerical advantage since each kernel can be evaluated in a subroutine and it is thus easy to change kernels.

2.9.2 SPH advantages and drawbacks

SPH method has several advantages on Eulerian methods:

- For some simulations, the construction of an efficient Eulerian mesh can be delicate. It is the case for instance when simulating a star explosion or a breaking wave. The main advantage of SPH is then the total absence of mesh. Moreover, boundaries can easily have several forms and considering a mobile wall is not problematic.

- Equations discretised in SPH formalism and numerical implementation are quite simple. It is not the case of other particle methods such as MAC method (see chapter 1).
- Thanks to SPH, complex problems incorporate complete physical phenomena can be investigated: supplementary equations concerning several physical problems (diffusion, chemical reactions, two phase flows, ...) are introduced and resolved in this formalism.
- In Eulerian codes, the convective term $(\underline{u}.\nabla)$ of Navier-Stokes equations causes lots of problems (it induces numerical diffusion when it is discretised), whereas the particular derivative present in SPH equations avoids them.
- As shown in [84], SPH is consistent with the basic laws of mechanics: for instance, the momentum equation can be directly derived from an action principle.

However, SPH has some drawbacks as well:

- Since most SPH codes use a completely explicit integration scheme, the time step is much smaller than the one relative to other numerical methods for incompressible flows, when using a weakly compressible algorithm involving a speed of sound c_0 (see part 2.4.3).
- The problem of numerical instabilities is until now not completely resolved (see part 2.8) and wall treatment can be delicate in some cases (see part 2.7).
- Turbulence modelling in SPH is until now very scarce. However, some recent tests are hopeful ([94]) and it is now possible to use mixing length [39], [93] and $k - \epsilon$ models for nearly incompressible flows [89]. Turbulence modelling trough SPH is an important part of this thesis.

2.9.3 Complementarity

SPH nodes have been linked to finite elements by Johnson et al [40]. This allows highly distorted materials to be considered together with structural response material. Indeed, although SPH approaches are very efficient for severe distorsions, they are generally not as good as standard finite elements for structural response applications [40]. Figure 2.24 shows two computations of a strong steel penetrator impacting a much softer aluminium target. The left picture is a linked computation with standard finite elements for the penetrator modelling and SPH nodes for the target modelling. The computed residual velocity is in good agreement with experimental results. This type of problem is ideally suited for linkage of SPH nodes and standard elements. The right picture of figure 2.24 shows the same test-case except that both the penetrator and target are modelled by SPH nodes. In this case, the penetrator is significantly deformed and it does not perforate the target plate. A long term objective could be to allow the user to define almost any impact problem with a standard finite element grid and then to allow the standard elements to be converted to SPH nodes as the standard elements become distorted. Although this approach has already been demonstrated, more work is required to increase the accuracy and robustness for a wider range of problems.

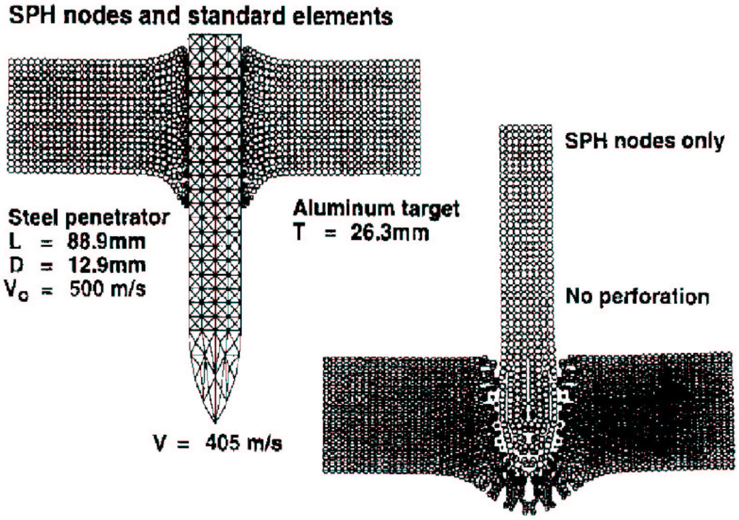

Figure 2.24: Penetration computations using SPH nodes linked to standard elements (left) and using only SPH nodes (right) [40].

Chapter 3

The numerical code Spartacus-2D

The starting point of this PhD work is the SPH code Spartacus-2D, being developed at the *"Laboratoire National d'Hydraulique et d'Environnement"* of EDF (*"Electricité de France"*) since 1998. The acronym SPARTACUS means "Smoothed PARTicle hydrodynamics for ACUrate flow Simulation". This 2D SPH code is equipped with pre and post processors and is documented for the user and the developer [85]. The code is very compact (less than 1 000 lines) and many of the routines were modified and extended during this work.
This chapter describes the fundamental equations implemented in Spartacus-2D and presents its general algorithm structure. The turbulence modelling implemented in Spartacus-2D will be discussed in chapter 5.

3.1 Fundamental equations

3.1.1 The continuity equation

As previously mentioned, the continuity equation can be written in several ways in SPH formalism (see part 2.4.1). In Spartacus-2D, only the expression (3.1)

$$\frac{d\rho_a}{dt} = \sum_b m_b \underline{u}_{ab} \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{3.1}$$

is used to compute the density of each particle at each time step.

3.1.2 The momentum equation

Pressure gradient

Two discretisations of the pressure gradient are implemented in Spartacus-2D:

$$\nabla p_a = \sum_b \frac{m_b}{\rho_b} \left(p_b + p_a\right) \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{3.2}$$

and

$$\nabla p_a = \rho_a \sum_b m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} \tag{3.3}$$

The viscous term

In order to create an efficient code, the viscous term (2.65) which directly stemms from the Lagrangian momentum equation is not implemented in Spartacus-2D. However, the viscous modelling developed by Monaghan

$$\nu \Delta \underline{u}_a = \sum_b \frac{16\nu}{\rho_a + \rho_b} \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \dot{w}_h \left(r_{ab} \right) \cdot \underline{\epsilon}_{ab} \tag{3.4}$$

is considered. As mentioned in the previous chapter, the kinematic viscosity ν can be replaced by $\frac{1}{2}\left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b}\right)$, which gives

$$\nu \Delta \underline{u}_a = \sum_b \frac{8}{\rho_a + \rho_b} \left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b} \right) \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \dot{w}_h \left(r_{ab} \right) \cdot \underline{\epsilon}_{ab} \tag{3.5}$$

In oder to validate the code for low Reynolds number flows, the viscous modelling achieved by Morris is also implemented:

$$\nu \Delta \underline{u}_a = \sum_b \frac{m_b (\mu_a + \mu_b) \underline{r}_{ab} \cdot \dot{w}_h (r_{ab}) \underline{\epsilon}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + \eta^2 \right)} \underline{u}_{ab} \tag{3.6}$$

Therefore, with equation (3.2) for the pressure gradient term, the two following forms of the momentum equation are considered in Spartacus-2D:

$$\frac{d\underline{u}_a}{dt} = - \sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8}{\rho_a + \rho_b} \left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b} \right) \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \underline{F}_a^e \tag{3.7}$$

$$\frac{d\underline{u}_a}{dt} = - \sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \sum_b \frac{m_b \left(\mu_a + \mu_b \right) \underline{r}_{ab} \cdot \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + \eta^2 \right)} \underline{u}_{ab} + \underline{F}_a^e \tag{3.8}$$

where $\underline{F}_a^e$ correspond to external forces applied on particle a.

3.1.3 External forces

Gravity effects

When gravity effects are considered, equations (3.7) and (3.8) respectively become

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8}{\rho_a + \rho_b} \left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b} \right) \frac{\underline{u}_{ab} . \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{e}_{ab} + \underline{g} \tag{3.9}$$

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} \right) \dot{w}_h \left(r_{ab} \right) \underline{e}_{ab} + \sum_b \frac{m_b \left(\mu_a + \mu_b \right) \underline{r}_{ab} . \dot{w}_h \left(r_{ab} \right) \underline{e}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + \eta^2 \right)} \underline{u}_{ab} + \underline{g} \tag{3.10}$$

Propelling force

In Spartacus-2D, there are two main ways to set a fluid in motion:

- As in finite-differences codes, the user can apply a constant force F to all fluid particles. This force is considered as an external volumetric force and added in the r.h.s. of the momentum equation (3.7) or (3.8).

- For a periodic flow, the forcing term F can also be updated at each time step to impose the correct mass flow rate at the "inlet". In finite volume codes [71], the same type of method is used: the flow is driven by a pressure gradient $\partial P / \partial x$ applied between the "inlet" and the "outlet". This pressure gradient is then updated at each time step by the following relaxation formula:

$$\frac{\partial P}{\partial x} = \frac{\partial P}{\partial x} + \frac{2 \left(\dot{m}_n - \dot{m}_{ref} \right) - \left(\dot{m}_{n-1} - \dot{m}_{ref} \right)}{2 \rho \delta t} \tag{3.11}$$

where $\dot{m}_n$ corresponds to the mass flow rate at the time n and $\dot{m}_{n-1}$ at the time $n-1$. $\dot{m}_{ref}$ denotes the reference mass flow rate which is imposed by the user and δt the time step. Mass flow rates are computed at the "inlet" according to

$$\dot{m} = \frac{1}{H} \int_0^H \rho \underline{u} . \underline{n} dz \tag{3.12}$$

where H denotes the "inlet" height, $\underline{u}$ the velocity vector and $\underline{n}$ the normal vector relative to the inlet section. In a 2D channel case, equation (3.12) becomes:

$$\dot{m} = \frac{1}{H} \int_0^H \rho u(z) dz \tag{3.13}$$

where $u(z)$ corresponds to the axial component of the velocity.
This relaxation of the driving pressure gradient has been adapted to SPH. At first, the

pressure gradient is expressed as a force per mass F according to

$$F = -\frac{1}{\rho}\frac{\partial P}{\partial x} \tag{3.14}$$

The relaxation equation (3.11) is thus written

$$F = F - \frac{1}{\rho}\frac{2\left(\dot{m}_n - \dot{m}_{ref}\right) - \left(\dot{m}_{n-1} - \dot{m}_{ref}\right)}{2\delta t} \tag{3.15}$$

In SPH formalism, the mass flow rate relative to the "inlet" of a 2D channel can be computed according to:

$$\dot{m} = \frac{1}{N_b}\sum_b \rho_b u_b \tag{3.16}$$

where b denotes all the particles located in a zone located between $x = x_{``inlet''}$ and $x = x_{``inlet''} + d$, as shown in figure 3.1. The distance d is arbitrarily chosen by the user.

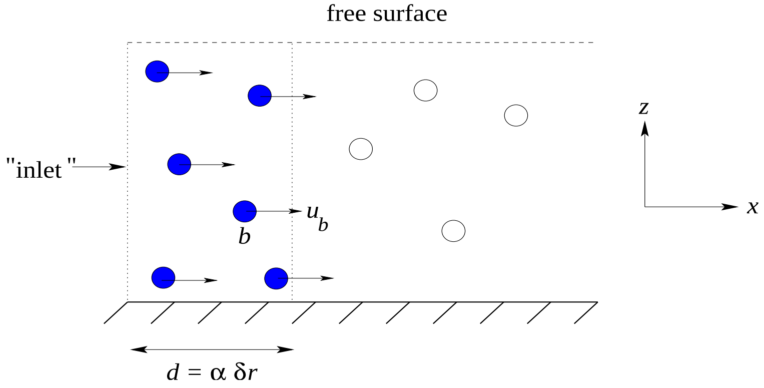

Figure 3.1: Computation of the mass flow rate at the "inlet" of a 2D chanel with SPH.

The value N_b in equation (3.16) corresponds to the number of particles located in the previous zone and u_b their axial velocity. By updating the propelling force at each time step, the CPU time required to reach an equilibrium state when simulating a stationary flow is much reduced.

Repulsive wall forces and wall modelling

Walls can be modelled with Spartacus-2D by considering repulsive forces: three types of repulsive central forces are implemented, among them the force $\underline{f}_1$ (see equation (2.113)) and the Lennard-Jones force $\underline{f}_2$ (see equation (2.114)). If a fluid particle a is linked to a wall particle b, the contribution of the repulsive force due to b is considered as an external volumic force and added in the r.h.s. of the momentum equation.

It is also possible to model walls with wall and mirror particles, as described in part 2.7.2. In this case, repulsive forces are not considered.

3.1.4 State equation

The state equation implemented in Spartacus-2D corresponds to equations (3.17)

$$p\left(\rho\right) = B\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{3.17}$$

where

$$B = \frac{\rho_0 c_0^2}{\gamma} \qquad \textit{and} \qquad \gamma = 7 \tag{3.18}$$

ρ_0 and c_0 respectively denote the reference density and a numerical speed of sound chosen by the user. When simulating a nearly incompressible flow, c_0 must be at least ten times superior to the maximal velocity of the flow (see part 2.5.1).

3.1.5 Particle position

In the current version of Spartacus-2D, only a simple velocity integration (see part 2.4.4) gives the new particle position according to:

$$\frac{d\underline{r}_a}{dt} = \underline{u}_a \tag{3.19}$$

XSPH method (see part 2.4.4) has been tested as well but gave no convincing results.

3.2 Kernels

Three spline kernels (third, fourth and fifth order ones) can be selected in Spartacus-2D. The definition of these kernels respectively correspond to equations (2.29), (2.30) and (2.31). Nevertheless, only their first derivative, respectively defined by equations (3.20), (3.21) and (3.22), is computed and used at each time step:

$$\dot{w}_h\left(q\right) = \frac{10}{7\pi h^2}\begin{cases} -3q + \frac{9}{4}q^2 & \text{if} \quad 0 \leq q \leq 1 \\ -\frac{3}{4}\left(2-q\right)^2 & \text{if} \quad 1 \leq q \leq 2 \\ 0 & \text{if} \quad q \geq 2 \end{cases} \tag{3.20}$$

$$\dot{w}_h\left(q\right) = \frac{51}{2000h^2}\begin{cases} -4\left(\frac{5}{2}-q\right)^3 + 20\left(\frac{3}{2}-q\right)^3 - 40\left(\frac{1}{2}-q\right)^3 & \text{if} \quad 0 \leq q \leq 0.5 \\ -4\left(\frac{5}{2}-q\right)^3 + 20\left(\frac{3}{2}-q\right)^3 & \text{if} \quad 0.5 \leq q \leq 1.5 \\ -4\left(\frac{5}{2}-q\right)^3 & \text{if} \quad 1.5 \leq q \leq 2.5 \\ 0 & \text{if} \quad q \geq 2.5 \end{cases} \tag{3.21}$$

$$\dot{w}_h(q) = \frac{7}{478\pi h^2} \begin{cases} -5(3-q)^4 + 30(2-q)^4 - 75(1-q)^4 & \text{if} \quad 0 \leq q \leq 1 \\ -5(3-q)^4 + 30(2-q)^4 & \text{if} \quad 1 \leq q \leq 2 \\ -5(3-q)^4 & \text{if} \quad q \geq 3 \\ 0 & \text{if} \quad q \geq 3 \end{cases} \tag{3.22}$$

where q denotes the ratio r_{ab}/h.

3.3 Time step and temporal integration

3.3.1 Time step

The time step δt is evaluated according to the three conditions presented in part (2.5.1), involving the forces condition, the viscous criterion and the CFL condition:

$$\delta t = \min\left(\delta t_{forces}, \delta t_{CFL}, \delta t_{visc}\right) \tag{3.23}$$

3.3.2 Temporal integration

The Navier-Stokes and the position equations can be integrated in time with the fully explicit method

$$\begin{cases} \underline{u}_a^{n+1} = \underline{u}_a^n + \delta t \underline{F}_a^n \\ \rho_a^{n+1} = \rho_a^n + \delta t \underline{G}_a^n \\ \underline{r}_a^{n+1} = \underline{r}_a^n + \delta t \underline{H}_a^{n+1} \end{cases} \tag{3.24}$$

or the predictor corrector scheme described in part (2.5.2). $\underline{F}$ and $\underline{G}$ respectively correspond to the r.h.s. of equations (3.7) or (3.8) and (3.1). $\underline{H}$ corresponds to the velocity $\underline{u}$.

3.4 Optimisation and algorithm structure

3.4.1 Link list

R.h.s. of the Navier-Stokes equations involving summations on each particle pair, the CPU time would scale as N^2 where N denotes the total particle number. However, since spline kernels have a compact support, each particle a is only linked to its closest neighbours b. It is thus important to construct an optimal link list relative to the particle connections at each time step: for each particle a, all its closest neighbours b will be given by the link list according to the following algorithm.

1. A coarse grid, composed of cells measuring h_t (see figure 3.2), is firstly applied to the fluid domain. We remind the reader that h_t is the support size of the kernel.

2. The corresponding cell of each particle a is determined.

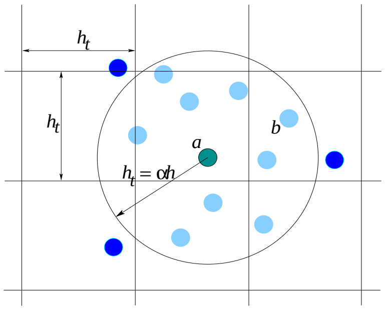

Figure 3.2: Determination of the closest neighbours relative to particle a.

3. For each particle a, particles b located in cell a and in the eight adjacent cells are considered. Only those for which $r_{ab} < h_t$ are considered as a-neighbours.

The use of the coarse grid considerably reduces the number of particles which are potentially susceptible to be considered as a-neighbours. Indeed, the test corresponding to $r_{ab} < h_t$ is thus not applied to all particles b present in the calculation domain. Hence, this algorithm remains proportional to N.

3.4.2 Periodic conditions

In Spartacus-2D, periodic conditions (with respect to x in figure 3.3) are achieved by considering that particles located at a boundary are linked to particles located at the opposite boundary. The cells adjacent to the a-particle one are therefore completed by those located in the vicinity of the opposite boundary. Consequently, the code considers that particle b of figure 3.3 is situated at $x_b - (x_{max} - x_{min})$ and particle b could then become a a-neighbour. It is the same for particles which are located in the vicinity of the right boundary.

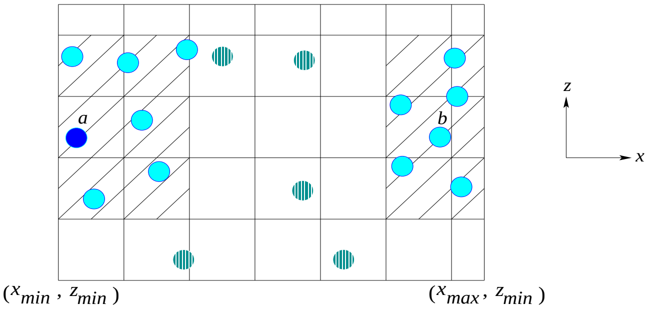

Figure 3.3: Adjacent cells relative to a periodic flow with respect to x-direction.

3.4.3 Asymmetrical principles

When constructing the link list, the pair of particles which interact is only counted once: due to the asymmetrical principle described in part 2.2.3, if a particle a interacts with a particle b, this last one automatically interacts with a. Consequently, only couples a, b checking $n_a < n_b$ are examined, n_i defining here the reference number of particle i. Thanks to this method, evaluation of all asymmetric terms present in the Navier-Stokes equations are efficiently computed. Besides reducing calculation time, this method reduces the memory by a factor 2.

3.4.4 Code structure

The simplified algorithm structure of Spartacus-2D is described in figure 3.4 and an example of results given by Spartacus2D is shown in figure 3.5.

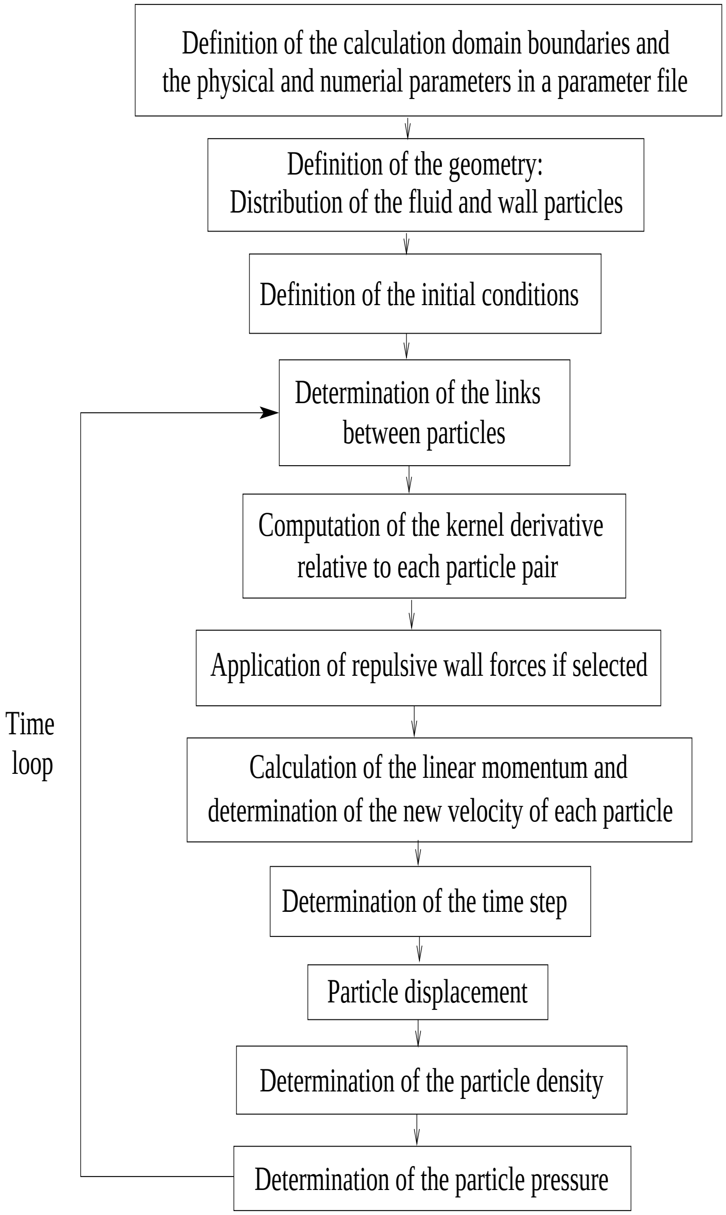
Definition of the calculation domain boundaries and the physical and numerial parameters in a parameter file
Definition of the geometry: Distribution of the fluid and wall particles
Definition of the initial conditions
Determination of the links between particles
Computation of the kernel derivative relative to each particle pair
Application of repulsive wall forces if selected
Calculation of the linear momentum and determination of the new velocity of each particle
Determination of the time step
Particle displacement
Determination of the particle density
Determination of the particle pressure
Time loop

Figure 3.4: Simplified algorithm structure of Spartacus-2D.

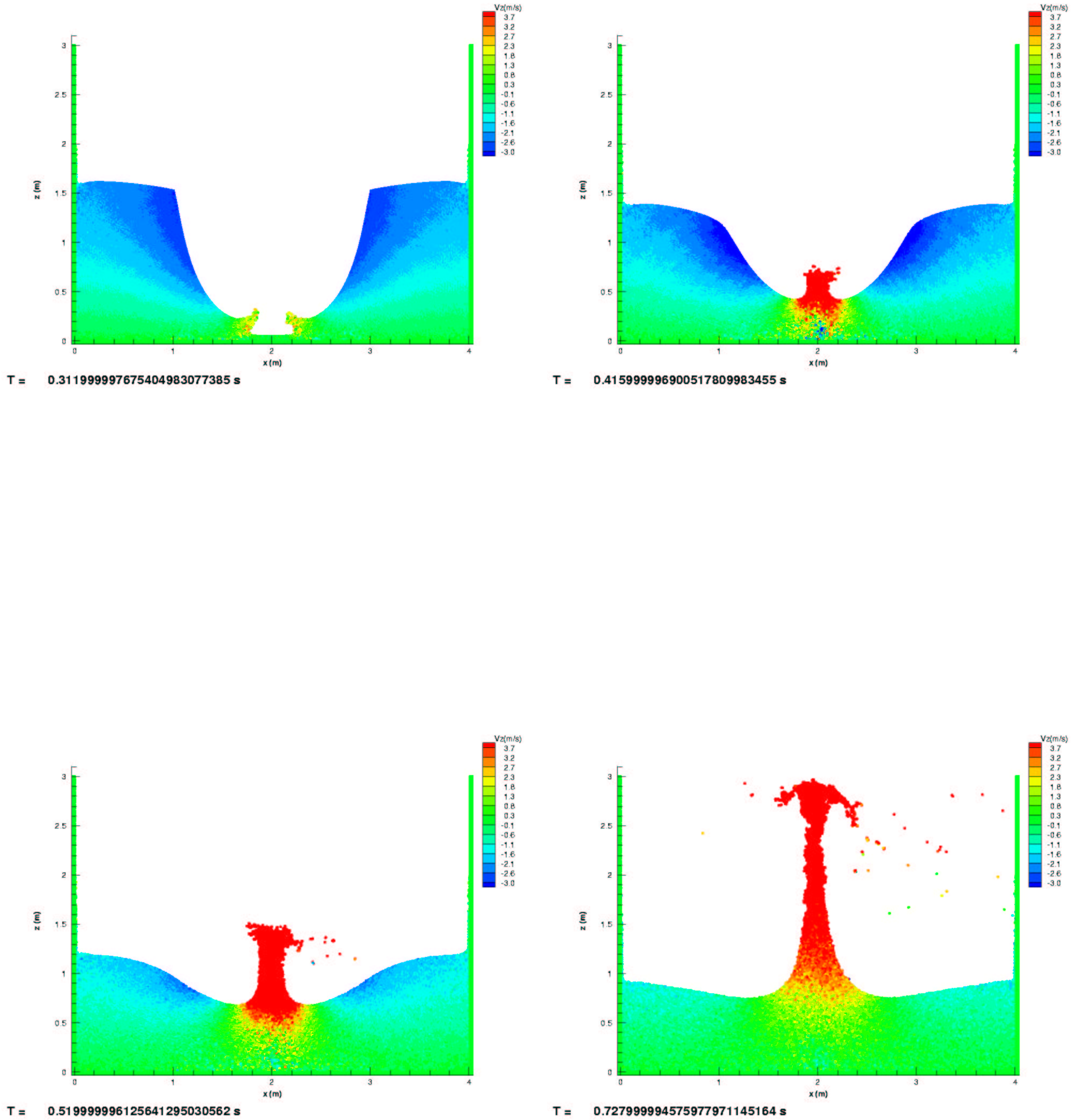

Figure 3.5: Vertical velocity field of two dam breaks by Spartacus-2D.

Chapter 4

Laminar incompressible flow simulations with Spartacus-2D

The study of laminar incompressible flows in which viscous forces are either comparable with or dominate inertial forces has various industrial and environmental applications. Indeed, many problems related to environment, mechanics, chemistry and petroleum engineering likewise involve slow viscous incompressible flows through filters, substrates, porous materials, and other potentially deformable structures. Although SPH is a versatile method, it is surprisingly difficult to test on simple laminar flows with precision [64] and the literature is quite scarce regarding SPH validation of this type of flows. This chapter presents some results of simulations of academic incompressible laminar flows with the SPH code Spartacus-2D.

4.1 Static phase in a 2D free surface channel[1]

The first test case corresponds to a static phase in a 2D free surface channel (see figure 4.1). The fluid is considered as a nearly incompressible one, as described in part 2.4.3. Contrary to Monaghan [59], we think that applying the theoretical density and pressure to fluid particles distributed on a regular lattice may not be sufficient. Indeed, since the fluid is weakly compressible and due to gravity effects, the particle position will slightly be modified: a real simulation of this static phase should therefore be done.

[1] Published in “Revue Europénne des Eléments Finis” journal & submitted to “Journal of Hydraulic Research”.

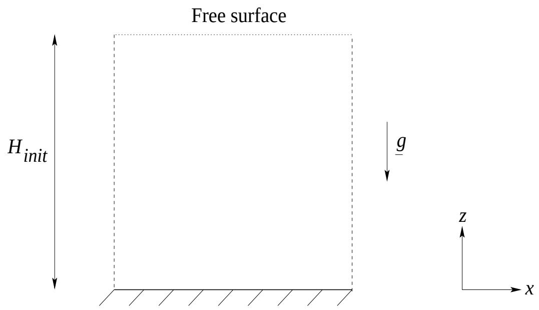

Figure 4.1: 2D free surface channel scheme.

4.1.1 Theoretical investigation

Theoretical density field

The projection of the fluid static law on the z-axis gives

$$-\rho g = \frac{\partial p}{\partial z} = \frac{dp}{d\rho}.\frac{d\rho}{dz} \tag{4.1}$$

By differentiating the state equation (3.17), it follows that

$$\frac{dp}{d\rho} = \frac{c_0^2}{\rho_0^{\gamma-1}}\rho^{\gamma-1} \tag{4.2}$$

By combining (4.1) and (4.2), one obtains

$$-\rho g = \frac{c_0^2}{\rho_0^{\gamma-1}}\rho^{\gamma-1}\frac{d\rho}{dz} \tag{4.3}$$

The origin of the z-axis is located at the bottom level. Integrating the relation (4.3) between the free surface altitude $z = H$ (after the settling phase) and the altitude z

$$\int_{z=H}^{z} -gdz = \frac{c_0^2}{\rho_0^{\gamma-1}}\int_{\rho_0}^{\rho}\rho^{\gamma-2}d\rho \tag{4.4}$$

gives

$$\rho\left(z\right) = \rho_0\left[\frac{g\left(H-z\right)\left(\gamma-1\right)}{c_0^2}+1\right]^{\frac{1}{\gamma-1}} \tag{4.5}$$

We remind that ρ_0 denotes the reference particle density, c_0 the speed of sound and H the fluid depth after settling. Since the fluid is slightly compressible, H is very close to the initial depth H_{init}.

Fluid depth after settling

The fluid depth H after settling is determined by considering that the total mass is conserved during the settling process, i.e.

$$\int_{z=0}^{H} \rho\left(z\right) dz = \int_{z=0}^{H_{init}} \rho_0 dz \tag{4.6}$$

The expression of H is then given by

$$H = \left[\left(\frac{H_{init}\gamma g}{c_0^2}+1\right)^{\frac{\gamma-1}{\gamma}}-1\right]\frac{c_0^2}{g\left(\gamma-1\right)} \tag{4.7}$$

Theoretical pressure field

The theoretical pressure field is directly obtained by considering the state equation (3.17) and the density field established according to relation (4.5):

$$p\left(z\right) = \frac{\rho_0 c_0^2}{\gamma}\left[\left(\frac{g\left(H-z\right)\left(\gamma-1\right)}{c_0^2}+1\right)^{\frac{\gamma}{\gamma-1}}-1\right] \tag{4.8}$$

When $\gamma \to \infty$ (rigorous incompressible case), expressions (4.5), (4.7) and (4.8) respectively become $\rho = \rho_0$, $H = H_{init}$ and $p\left(z\right) = \rho_0 g\left(H-z\right)$ (purely hydrostatic case).

4.1.2 System modelling

Fluid discretisation and wall modelling

The 2D free surface channel is modelled according to figure 4.2, with periodic conditions applied in x-direction. The fluid discretisation is described in table 4.1: (40×40) (x, z) fluid particles are initially distributed on a regular lattice and spaced by a distance $\delta r = 1$ cm. The wall is modelled by 40 fixed wall particles and its impermeability is ensured by the mirror particle concept described in part 2.7.2: three layers of mirror particles are considered under the wall (see the zoom in figure 4.2), so that the fluid particles located close to the wall do not suffer from a lack of neighbours. As boundary conditions, their pressure and density evolve at each time step as described in part 2.7.2, and they repel fluid particles from the wall through the pressure gradient term. In this case, mirror particles are characterised by a zero velocity and are involved in the viscous term evaluation.

As initial conditions, all the particles are initially characterised by a zero velocity and a reference density ρ_0.

Fluid particles (x, z)	40×40
Wall particles	40
Mirror particles	120
Total particle number	1 760
Particle initial spacing: δr (cm)	1.

Table 4.1: Fluid, wall and mirror particle discretisation for the settling phase simulation.

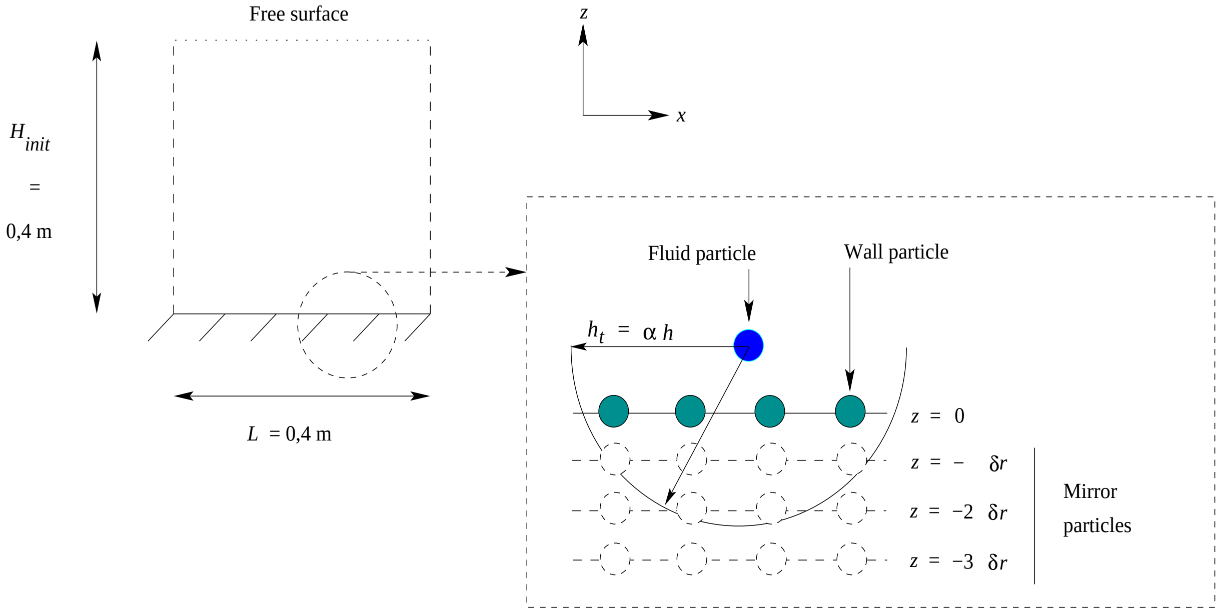

Figure 4.2: 2D free surface channel modelling (left) and zoom near the wall (right).

Order of the spline kernel	3
$h/\delta r$	1.5
$c_0\ \left(m.s^{-1}\right)$	30

Table 4.2: Main SPH numerical parameters for the settling phase simulation.

SPH equations and numerical parameters

Since the continuity equation (3.1) is used to compute the fluid particle density, the pressure gradient of the momentum equation is discretised according to equation (3.2), in order to respect the variational consistent (see part 2.4.2). Viscous effects are modelled by Monaghan's expression (3.4) and the gravity $\underline{g}$ is considered as an external force. The momentum equation for particle a is therefore written

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8\nu}{\rho_a + \rho_b} \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab}\right) \underline{e}_{ab} + \underline{g} \tag{4.9}$$

The Navier-Stokes and the position equations are integrated in time with the fully explicit method described in part 3.3.2. The pressure of the particles is then determined through the state equation (3.17). The main SPH numerical parameters for this test case are presented in table 4.2. If one considers that the maximal velocity of this flow is inferior to $\sqrt{gH_{init}}$, a suitable value of the speed of sound is approximately $10 \times \sqrt{gH_{init}}$, i.e. 20 $m.s^{-1}$.

4.1.3 Simulation results

While simulating the settling phase, numerical instabilities appear near the wall around $t = 1.8$ s: figure 4.3 shows that particles are disorganised and clump into several groups. These perturbations grow with time and propagate towards the free surface. A numerical checking of the criterion (2.119) [35] proved that these perturbations are identical to tensile instabilities described in part 2.8.1. To obtain a real steady state, the damping term described by Monaghan (see part 2.6.2) is introduced in the momentum equation in order to dissipate the residual kinetic energy. The new momentum equation for particle a then becomes

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8}{\rho_a + \rho_b} \left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b} \right) \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab}\right) \underline{e}_{ab} + \underline{g} - \Gamma \underline{u}_a \tag{4.10}$$

The numerical damping term is characterised by a coefficient Γ which has to be large enough and to not perturb the calculations: numerical tests revealed that the value $\Gamma \approx 10\ 000\ s^{-1}$ is suitable and it also satisfies $\Gamma|\underline{u}_a| << |\underline{g}|$. Moreover, this value is consistent with the one found by Monaghan in [57]. As the instabilities firstly appear in the vicinity of the wall, we first apply this damping term in the zone $z < 0.1$ m. The left picture of figure 4.4 reveals that the numerical instabilities travel through this zone without disorganising the fluid particles.

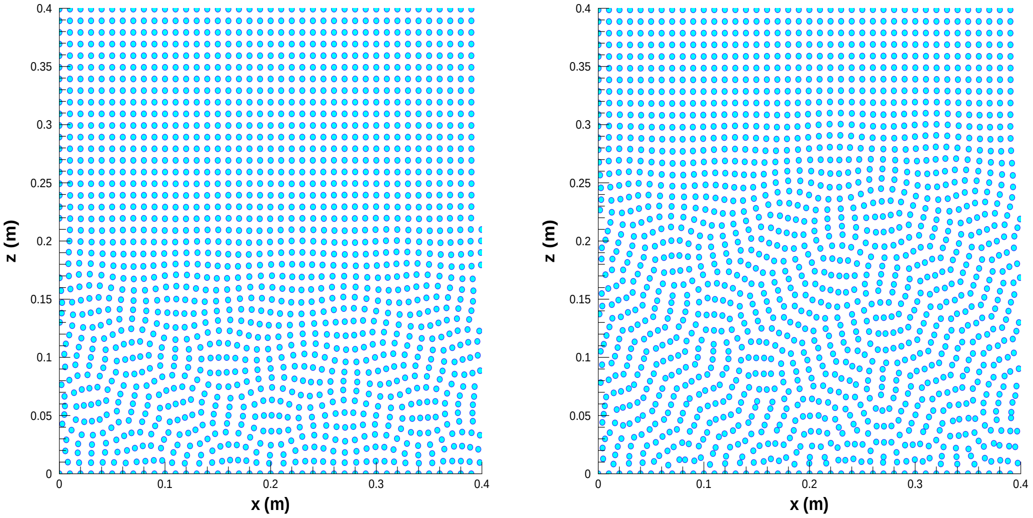

Figure 4.3: Particle disorganisation at $t = 1.5$ s (left) and $t = 1.8$ s (right).

	Theory	Spartacus-2D	Relative error in %
H (m)	0.399687	0.399676	$2,8.10^{-3}$

Table 4.3: Theoretical and Spartacus-2D evaluations of the fluid depth after settling.

Nevertheless, they reappear above $z = 0.1$ m. When applying this term to the entire fluid domain, particles remain organised, as shown on the right picture of figure 4.4. The right picture of figure 4.5 also reveals that the pressure field after 12 s is stable and satisfactory. The pressure profile presented on the same figure is obtained by spatially averaging the pressure field according to x-direction: one can notice that there is a close agreement between theoretical and Spartacus-2D results. Table 4.3 also shows that the fluid depth H after the settling phase is very close to the theoretical value.

We noticed that the use of a fourth order spline kernel cancels numerical instabilities, even if the damping term is removed. This result is consistent with the work of Morris [64] who reveals that a higher order kernel reduces numerical instabilities. However, the pressure profile is not as accurate as the one obtained with a third order spline kernel and a damping term (see [35]). This means that the introduction of the damping term is necessary to completely suppress the tensile instabilities during a settling process simulation. Nevertheless, the damping term is obviously not suitable for a dynamic simulation.

Considering finer spatial discretisations ((80 × 80) and (160 × 160)) does not avoid the propagation of numerical instabilities. Nevertheless, it increases the accuracy of the results, as shown on the pressure profiles of figure 4.6.

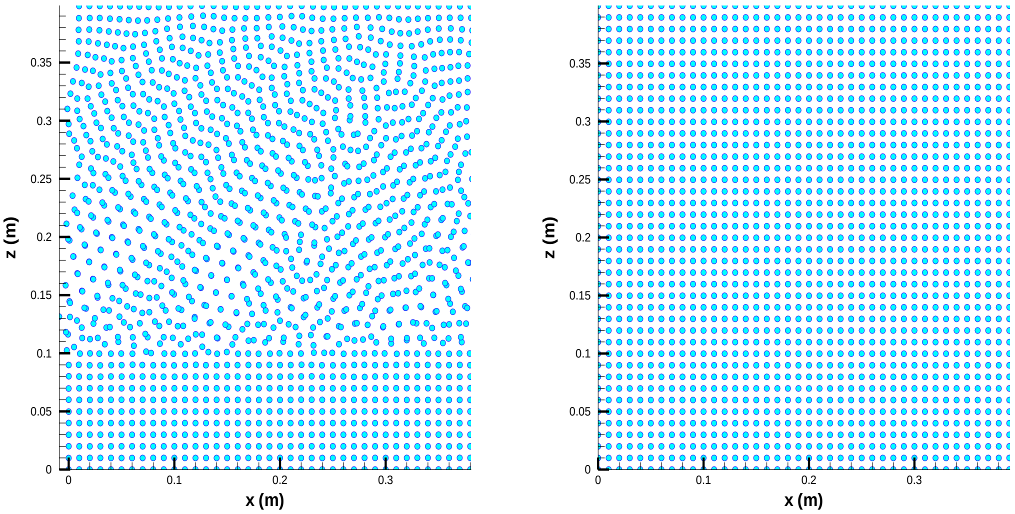

Figure 4.4: Particle position at $t = 8.5$ s with the application of a damping term in the zone $z < 0.1$ m (left) and at $t = 12$ s with the application of a damping term in the entire domain (right).

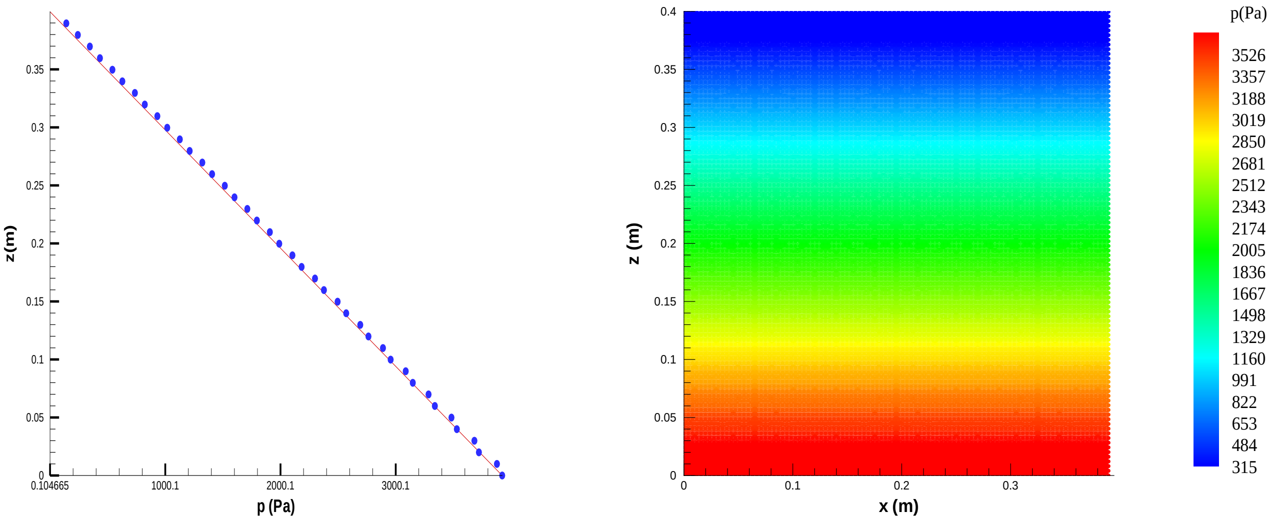

Figure 4.5: Pressure profile (left) and pressure field interpolated on a regular grid (right) after 12 s. —— : Theory; ●: Spartacus-2D.

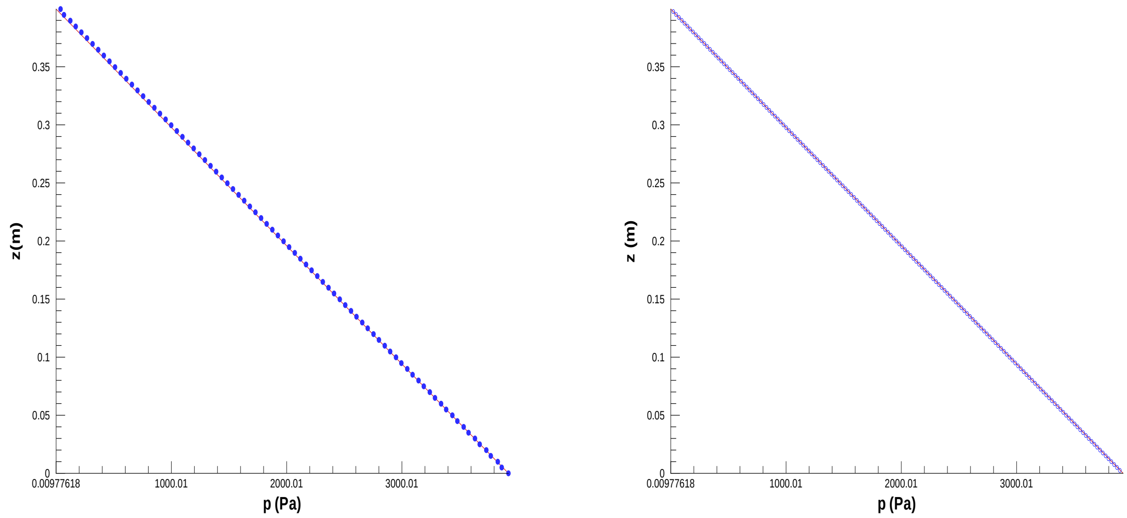

Figure 4.6: Pressure profile with a (80 × 80) (left) and a (160 × 160) (right) spatial discretisation. ——— : Theory; •: Spartacus-2D.

4.1.4 Partial conclusion

- A settling phase of a 2D free surface channel has been successfully simulated with the SPH code Spartacus-2D: the pressure profile is stable and very close to the theoretical one.

- The tensile instability was removed by introducing a damping term.

- The fourth order spline kernel also removed the tensile intability. This solution will be retained in the subsequent sections.

4.2 Laminar flow in a 2D free surface channel[2]

In [64], Morris explained that the simulation of a laminar Poiseuille flow with SPH is not so easy. In order to validate Spartacus-2D for this type of academic flows, the second test case corresponds to a laminar flow in the previous 2D free surface channel.

[2] Published in “Revue Europénne des Eléments Finis” journal & submitted to “Journal of Hydraulic Research”.

4.2.1 Theoretical investigation

Theoretical velocity field

We assume here that the spanwise component of the velocity field is zero. Hence, the continuity equation is written

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial z}=0 \tag{4.11}$$

where u and v respectively denote the axial (according to x) and the vertical (according to z) velocity component. Considering an invariance of the system in x-direction implies

$$\frac{\partial v}{\partial z}=0 \tag{4.12}$$

Moreover, since no slip-conditions require a zero velocity at the wall, it follows that $v=0$ in the entire domain.
The projection of the momentum equation on the x-axis gives

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial z^2}\right]+F^e \tag{4.13}$$

where p and ν respectively correspond to the pressure and the kinematic viscosity of the fluid. F^e denotes an external force per unit mass necessary to drive the fluid. For a stationary flow, the temporal term $\partial u/\partial t$ vanishes. Moreover, by considering that the system is invariant in x and that $v=0$, one obtains:

$$0=\nu\frac{\partial^2 u}{\partial z^2}+F^e \tag{4.14}$$

The z-integration of (4.14) hence gives

$$u\left(z\right)=-\frac{F^e}{2\nu}z^2+Az+B \tag{4.15}$$

where integration constants A and B are determined with the no-slip conditions at the wall and by considering that the viscous stress vanishes at the free surface. Therefore,

$$u\left(z\right)=\frac{F^e z}{2\nu}\left(2H-z\right) \tag{4.16}$$

Reynolds number

The Reynolds number is here defined by

$$Re=\frac{UH}{\nu} \tag{4.17}$$

Reynolds number: Re	10
Mean bulk velocity: U $(m.s^{-1})$	0.25
Fluid depth: H (m)	0.399676

Table 4.4: Physical characteristics of the free surface flow.

where H corresponds to the fluid depth after a settling phase and U the mean bulk velocity defined by

$$U = \frac{1}{H}\int_0^H u\left(z\right)dz = \frac{F^e H^2}{3\nu} \tag{4.18}$$

Wall viscous stress

The wall viscous stress τ is defined by

$$\tau = \mu \left.\frac{du}{dz}\right|_{z=0} \tag{4.19}$$

By considering relation (4.16), it follows that

$$\tau = \rho_{(z=0)} F^e H \tag{4.20}$$

4.2.2 Simulation conditions and system modelling

The physical characteristics of the simulated flow are described in table 4.4. The fluid is driven by a horizontal forcing term $\underline{F}^e$ applied to each fluid particle at each time step. This external force is constant in space but not in time since it is updated at each time step to impose the correct mass flow rate, as described in part 3.1.3. The standard system modelling and the numerical parameters for the previous test case are conserved, except the damping term which is removed. The momentum equation for particle a is thus written:

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8}{\rho_a + \rho_b}\left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b}\right)\frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2}\right)\dot{w}_h\left(r_{ab}\right)\underline{\epsilon}_{ab} + \underline{F}_a^e + \underline{g} \tag{4.21}$$

Moreover, to ensure particle stability, a fourth order spline kernel is used. Indeed, we checked in [35] that the use of a third order spline kernel results in tensile instabilities. The initial particle distribution corresponds to the (40×40) steady state obtained in part 4.1 and in order to reduce computing time, fluid particles are initially characterised by the following linear profile:

$$u\left(z\right) = \frac{F^e H}{2\nu} z \tag{4.22}$$

4.2.3 Simulation results

Results relative to the standard modelling

This part describes the results obtained with the standard modelling described in section 4.1.2. Figure 4.7 reveals that the axial velocity field is very stable: indeed, the particles remain organised and one can notice that the system is perfectly homogeneous along the x-direction. The axial velocity profiles presented in the following are obtained with a spatial averaging along x-direction. Figure 4.8 shows four axial velocity profiles at different times. Firstly, it proves that the code converges between 36 s and 48 s towards a stable velocity profile. Secondly, it reveals that the asymptotic axial velocity profile is overestimated by 6.6 % with the standard modelling. As we numerically checked that the residual pressure gradient according to x-direction was zero, a balance between the viscous forces and the forcing term $\underline{F}^e$ is established. Hence, the axial velocity overestimation is due to an underestimation of the viscous term, which has already been noticed by Violeau [93]. Morris [64] and Speith [76] also revealed that Monaghan's viscous term formulation (3.4) underestimates viscous effects when they are significant, as it is the case for the present low Reynolds number flow.

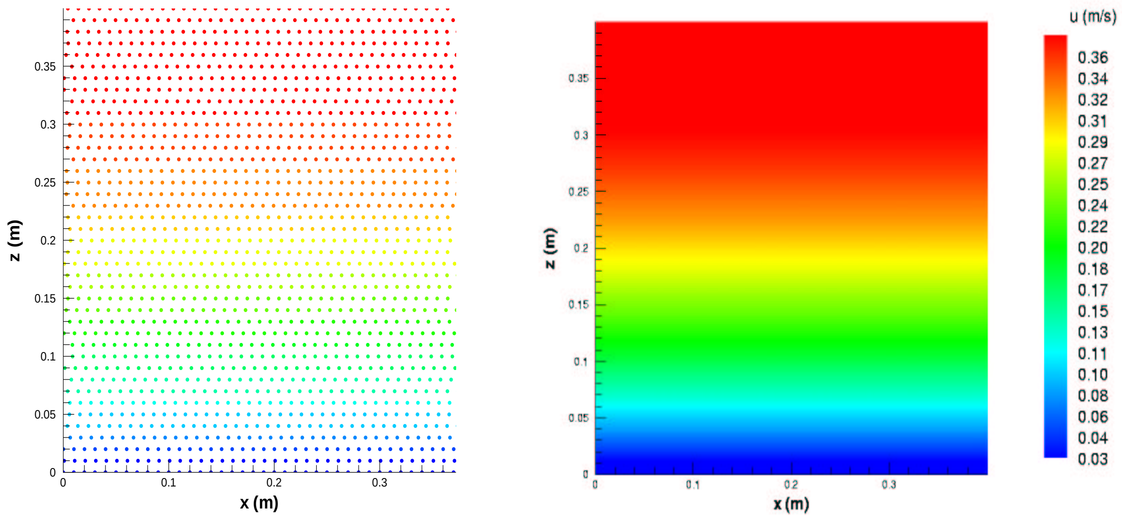

Figure 4.7: Particle position (left) and axial velocity field (right) with the standard modelling of the free surface flow.

Improved modelling

In order to reduce the previous axial velocity overestimation, the influence of several fundamental SPH parameters has been tested: Monaghan's formula (3.4) is at first compared with Morris' formula (3.6). The first one models viscous effects through normal forces between par-

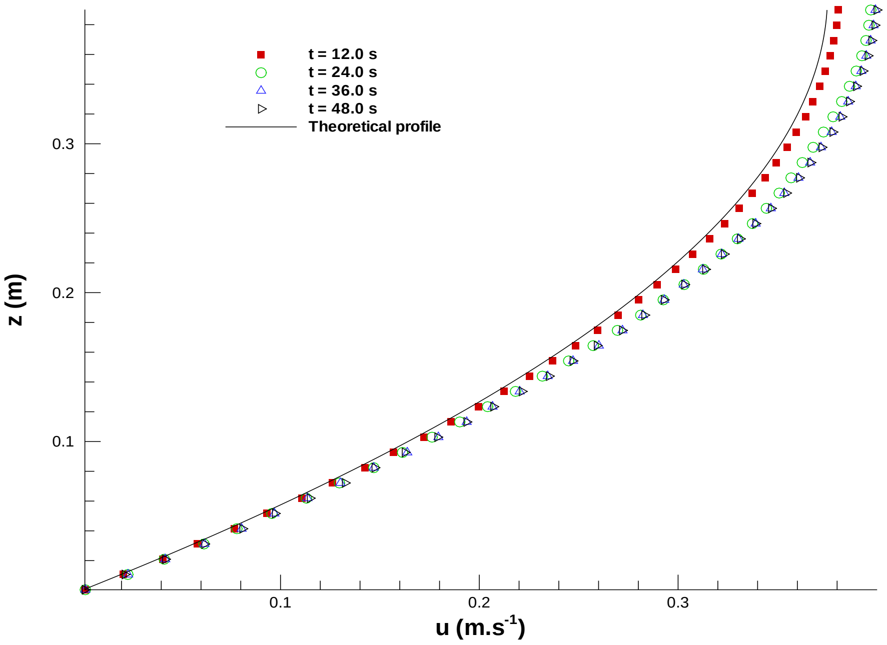

Figure 4.8: Axial velocity profiles at different times with the standard modelling of the free surface flow.

ticles whereas the second one considers a more physical modelling through tangential forces. The new momentum equation for particle a then becomes:

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b}\right) \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab} + \sum_b \frac{m_b \left(\mu_a + \mu_b\right) \underline{r}_{ab} . \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + 0.01h^2\right)} \underline{u}_{ab} + \underline{F}_a^e + \underline{g} \quad (4.23)$$

Then, to increase the wall viscous effects, mirror particles are characterised by a negative axial velocity, defined as the opposite of the symmetric fluid particle velocity (see figure 4.9). Velocity differences between mirror and fluid particles are consequently increased and the viscous term is thus larger (see equation (4.23)). This correction was also described by Morris in [64] in order to impose a real no-slip condition at the wall. As also explained in part 2.3.6, each SPH simulation is characterised by an optimal $h/\delta r$ value. By considering the wall viscous stress expression, Violeau [93] theoretically proved that the optimal $h/\delta r$ value for the flow considered here should be equal to 1.63. We numerically checked that this optimal value gave the best results, and we also noticed that particles clump into pairs when $h/\delta r$ exceeds 1.63. Since the quality of results depends on the number of particles as well, one can also increase the total number of particles. Therefore, a (80 × 80) spatial discretisation is considered. The results of the above mentioned options are presented in table 4.5: it reveals that the optimal parameters correspond to the use of Morris' formula to model viscous effects. Through mirror particles characterized by a symmetric negative velocity, the optimal $h/\delta r$ value equals 1.63 and a spatial discretisation with (80 × 80) particles. The axial velocity profile is then much more satisfactory (see figure 4.10) and presents an error of 0.2 %.

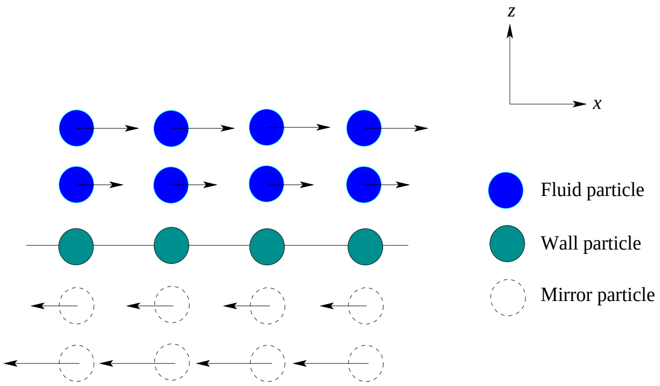

Figure 4.9: Mirror particles characterised by a symmetric negative axial velocity for the free surface flow.

4.2.4 Partial conclusion

With the standard SPH modelling, Spartacus-2D overestimates the velocity field of a laminar 2D free surface channel. However, by improving the viscous effects modelling and increasing the number of neighbours of each fluid particle, the velocity field is in very good agreement

	Viscous modelling	Mirror particles	$h/\delta r$	fluid discretisation	Relative error
Case 1	Monaghan (3.4)	Fixed	1.5	(40×40)	6.6 %
Case 2	Monaghan (3.4)	Mobile	1.5	(40×40)	6.2 %
Case 3	Monaghan (3.4)	Mobile	1.63	(40×40)	4.5 %
Case 4	Monaghan (3.4)	Mobile	1.63	(80×80)	3.5 %
Case 5	Morris (3.6)	Fixed	1.5	(40×40)	3.2 %
Case 6	Morris (3.6)	Mobile	1.5	(40×40)	2.6 %
Case 7	Morris (3.6)	Mobile	1.63	(40×40)	1.3 %
Best case	Morris (3.6)	Mobile	1.63	(80×80)	0.2 %

Table 4.5: Relative error on the axial velocity profile when modifying several fundamental SPH parameters for the free surface flow.

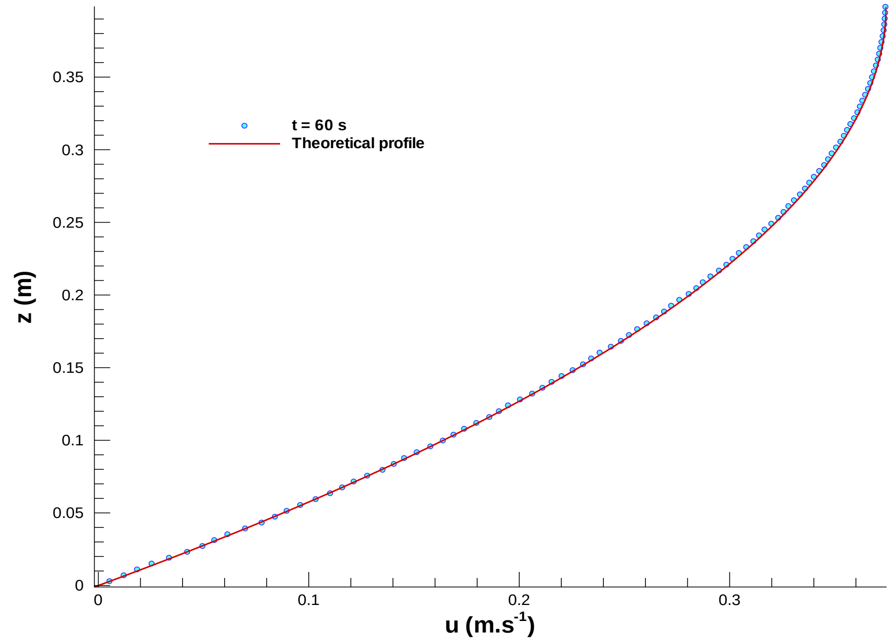

Figure 4.10: Axial velocity profile with the improved modelling of the free surface flow.

H (mm)	84.98
L (mm)	252
l (mm)	54
h (mm)	28
h_1 (mm)	56.98

Table 4.6: Dimensions of the 2D hill channel.

with the theoretical solution. This second test case validates Spartacus-2D and in a more general way SPH for the simulation of a laminar flow in a 2D free surface channel.

4.3 Laminar flow in a 2D hill channel[3]

The aim of this part is to reveal the ability of Spartacus-2D to reproduce recirculation zones. This third test case hence corresponds to an incompressible laminar flow in a 2D periodic hill channel. Contrary to the two previous test cases, a two wall bounded channel is here considered, without any gravity effect.

4.3.1 Geometry of the system

The geometry of the 2D hill channel corresponds to the one defined in ERCOFTAC workshops [81]: it is represented in figure 4.11 and defined in table 4.6. The hill is defined with 6 splines

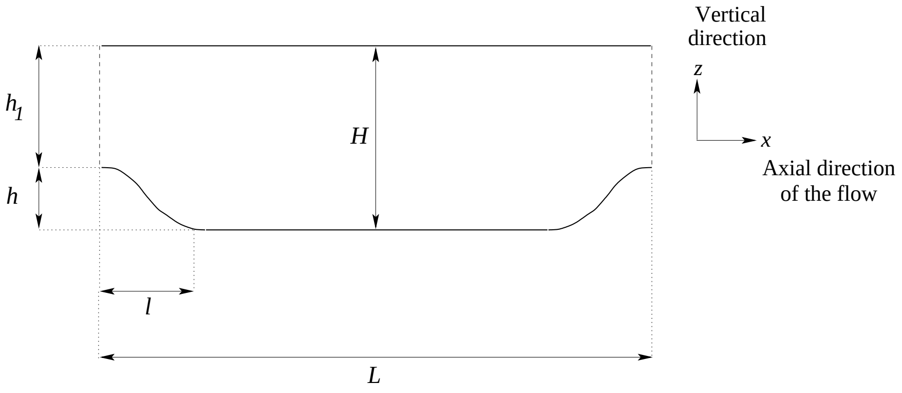

Figure 4.11: Geometry of the 2D hill channel.

[3] Presented at "Institute for Computational Fluid Dynamics" conference (Oxford, 2004) & to be published in a special issue of "International Journal for Numerical Methods in Fluids".

Reynolds number : Re	50
Mean bulk velocity : U $(m.s^{-1})$	$1,79.10^{-3}$

Table 4.7: Physical characteristics of the hill flow.

Fluid particles	19 548
Wall particles	502
Mirror particles	1 506
Total particle number	21 556
Particle initial spacing: δr (mm)	1.

Table 4.8: Fluid, wall and mirror particle discretisation for the hill flow.

according to the equation set (4.24):

$$z(x) = \begin{cases} 28 + 6,775.10^{-3}x^2 - 2,125.10^{-3}x^{-3} & \text{if} \quad 0 < x < 9 \\ 25.074 + 9,755.10^{-1}x - 1,016.10^{-1}x^2 + 1,889.10^{-3}x^3 & \text{if} \quad 9 < x < 14 \\ 25.796 + 8,207.10^{-1}x - 9,055.10^{-2}x^2 + 1,627.10^{-3}x^3 & \text{if} \quad 14 < x < 20 \\ 40.464 - x + 1,946.10^{-2}x^2 - 2,070.10^{-4}x^3 & \text{if} \quad 20 < x < 30 \\ 17.925 + 8,744.10^{-1}x - 5,567.10^{-2}x^2 + 6,277.10^{-4}x^3 & \text{if} \quad 30 < x < 40 \\ 56.390 - x + 1,645.10^{-2}x^2 + 2,675.10^{-5}x^3 & \text{if} \quad 40 < x < 54 \end{cases} \tag{4.24}$$

x and z respectively denote the axial and vertical position in millimeters.

4.3.2 System modelling

Simulation conditions

The physical characteristics of the laminar 2D hill flow are presented in table 4.7, where the Reynolds number is defined by

$$Re = \frac{Uh}{\nu} \tag{4.25}$$

U corresponds to the mean bulk velocity, h the hill height and ν the kinematic viscosity of the fluid.

Fluid discretisation and wall modelling

The fluid discretisation is described in table 4.8. The wall is still modelled with wall particles and three layers of mirror particles, as done for the previous test cases. The particles are initially distributed on a regular lattice, which may be an advantage comparing to Eulerian codes: indeed, it will be shown that there is no need to refine the distribution near the wall

Kernel order	4
$h/\delta r$	1.2
Speed of sound : c_0 $(m.s^{-1})$	0.03

Table 4.9: Main SPH numerical parameters for the hill flow.

with SPH. However, there at least two ways to refine a particle distribution with SPH: the first one is to consider a kernel with an adaptive support, which will consider more neighbours near the walls for instance. The second is to add particles in the zones which require refinement [43]. Due to the absence of gravity effect, there is no need to simulate a settling phase for this test-case. As boundary conditions, the mirror particle velocities do not evolve during the calculation (for simplicity reasons) but they contribute to the viscous term of fluid particles. In order to represent a periodic hill channel, periodic conditions are applied with respect to x-direction. As initial conditions, all the particles are initially characterised by a zero velocity and the reference density.

SPH equations and numerical parameters

As previously done, the continuity equation (3.1) and the pressure gradient term (3.2) are used for this third test-case. However, since the viscous effects are not consequent enough [36], the viscous term is modelled according to Monaghan's fomula (3.4). The gravity is here neglected and as previously, the fluid is driven by an external horizontal force $\underline{F}^e$ updated at each time step in order to impose the correct mass flow rate. The momentum equation for particle a hence becomes

$$\frac{d\underline{u}_a}{dt} = -\sum_b m_b \left(\frac{p_a + p_b}{\rho_a \rho_b} - \frac{8}{\rho_a + \rho_b} \left(\frac{\mu_a}{\rho_a} + \frac{\mu_b}{\rho_b} \right) \frac{\underline{u}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \underline{F}_a^e \qquad (4.26)$$

As the previous test cases, the fully explicit method is used to integrate the Navier-Stokes equations and the particle position. The fluid is still modelled with the nearly incompressible assumption and the particle pressure is therefore determined with the classical state equation (3.17). The main SPH numerical parameters are presented in table 4.9: the smoothing length value has been determined through numerical tests and the speed of sound is approximately equal to ten times the maximum velocity of the flow.

4.3.3 Simulation results

Since there is no theoretical solution for this problem, the results obtained with Spartacus-2D are compared to those obtained with the Eulerian code *Saturne*, based on finite volume method [81]. Firstly, figure 4.12 reveals that the axial and vertical velocity fields are on the whole satisfactory compared to Saturne's results. Moreover, the recirculation zone is quite

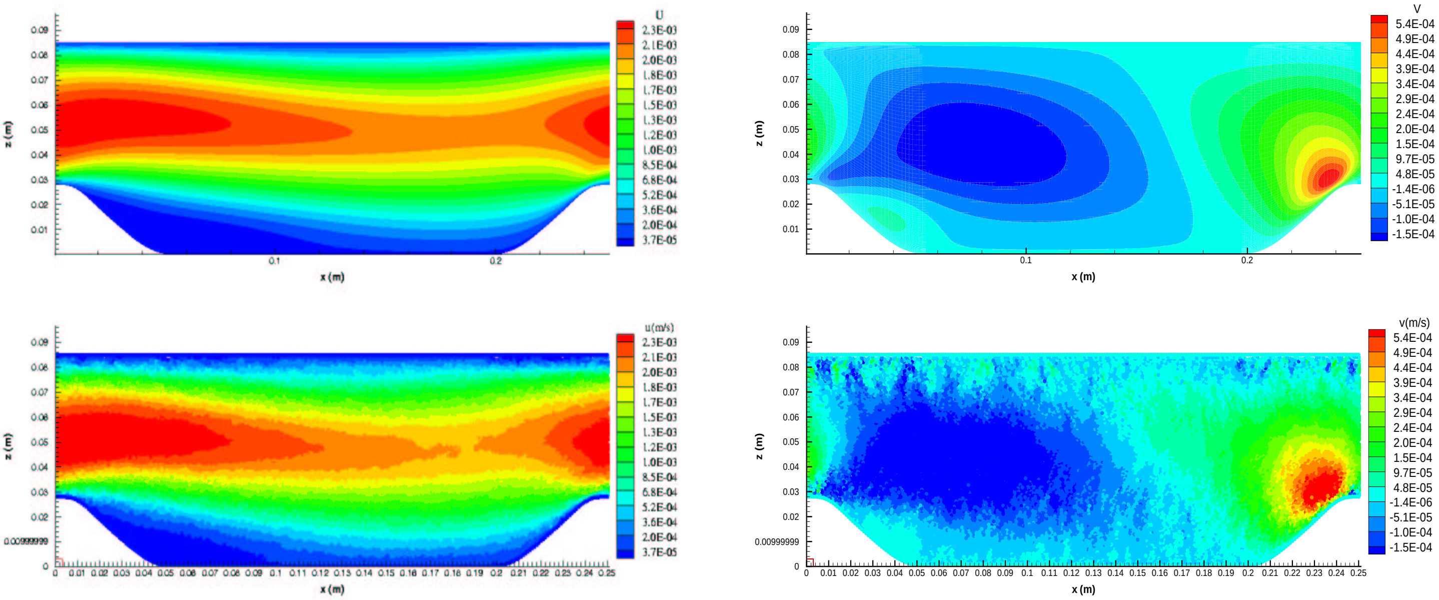

Figure 4.12: Axial (left) and vertical (right) velocity fields with Saturne (top) and Spartacus-2D (bottom) for the hill flow.

Profiles	x/h
P_1	0.05
P_2	1
P_3	2
P_4	3
P_5	5
P_6	8

Table 4.10: Position of the six considered profiles for the hill flow.

clear, as shown in figure 4.13. However, the vertical velocity field obtained with Spartacus-2D is noised, as shown on the right picture of figure 4.12. These perturbations will be smoothed in the following by a temporal average. In order to have more accurate comparisons, axial velocity profiles obtained with the two codes are considered. Six of the ten profiles defined in ERCOFTAC workshops [81] are used: they are shown in figure 4.14 and located in table 4.10. Figure 4.15 reveals that the axial velocity profiles obtained with Spartacus-2D are relatively close to those obtained with Saturne. Moreover, the profiles located in the vicinity of the hill (P_2, P_3 and P_4) prove the presence of the recirculation zone. Table 4.11 summarises the characteristics of this zone: the separation point is very well predicted with Spartacus-2D. Concerning the reattachment point, the error is more significant. Indeed, due to the Lagrangian characteristic of SPH, the determination of this length is quite delicate and only statistical. To smooth the numerical instabilities, a temporal average is applied after con-

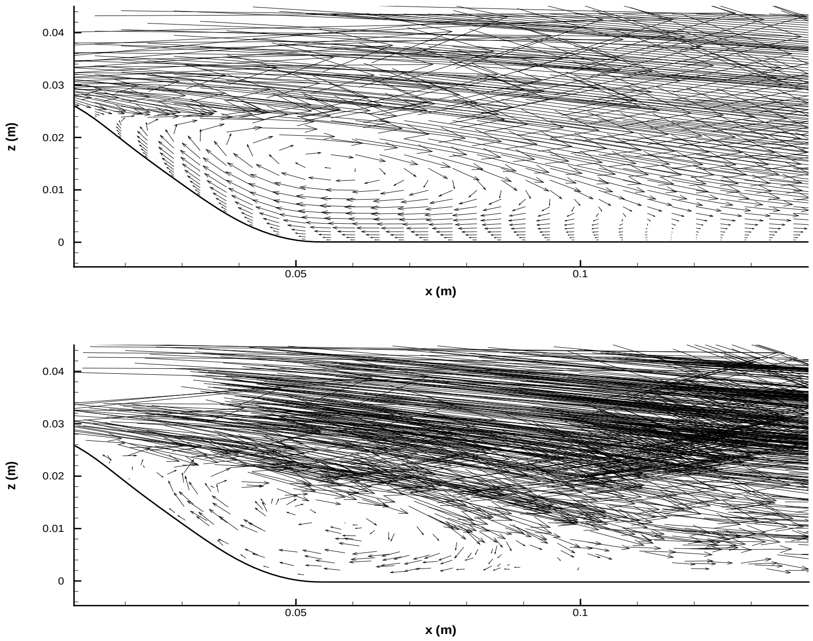

Figure 4.13: Velocity vectors in the vicinity of the downhill side with Saturne (top) and Spartacus-2D (bottom) for the hill flow.

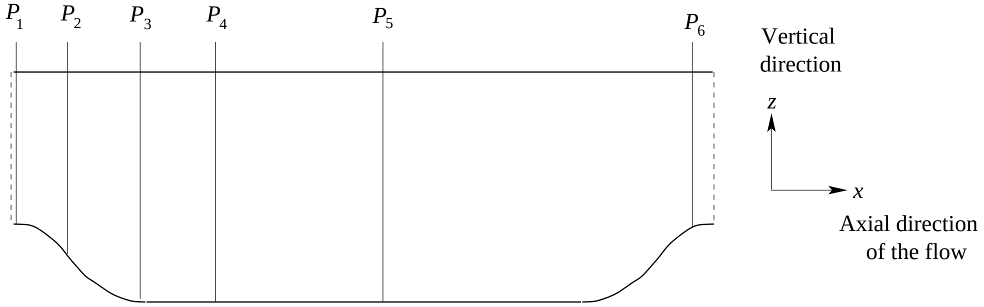

Figure 4.14: Position of the six profiles defined in ERCOFTAC project [81] and used herein for the validation of Spartacus-2D.

	Separation point (m)	Reattachment point (m)
Saturne	0.0147	0.120
Spartacus-2D	0.0146	0.115
Relative error (%)	0.7	4.3

Table 4.11: Separation and reattachment point prediction for the hill flow.

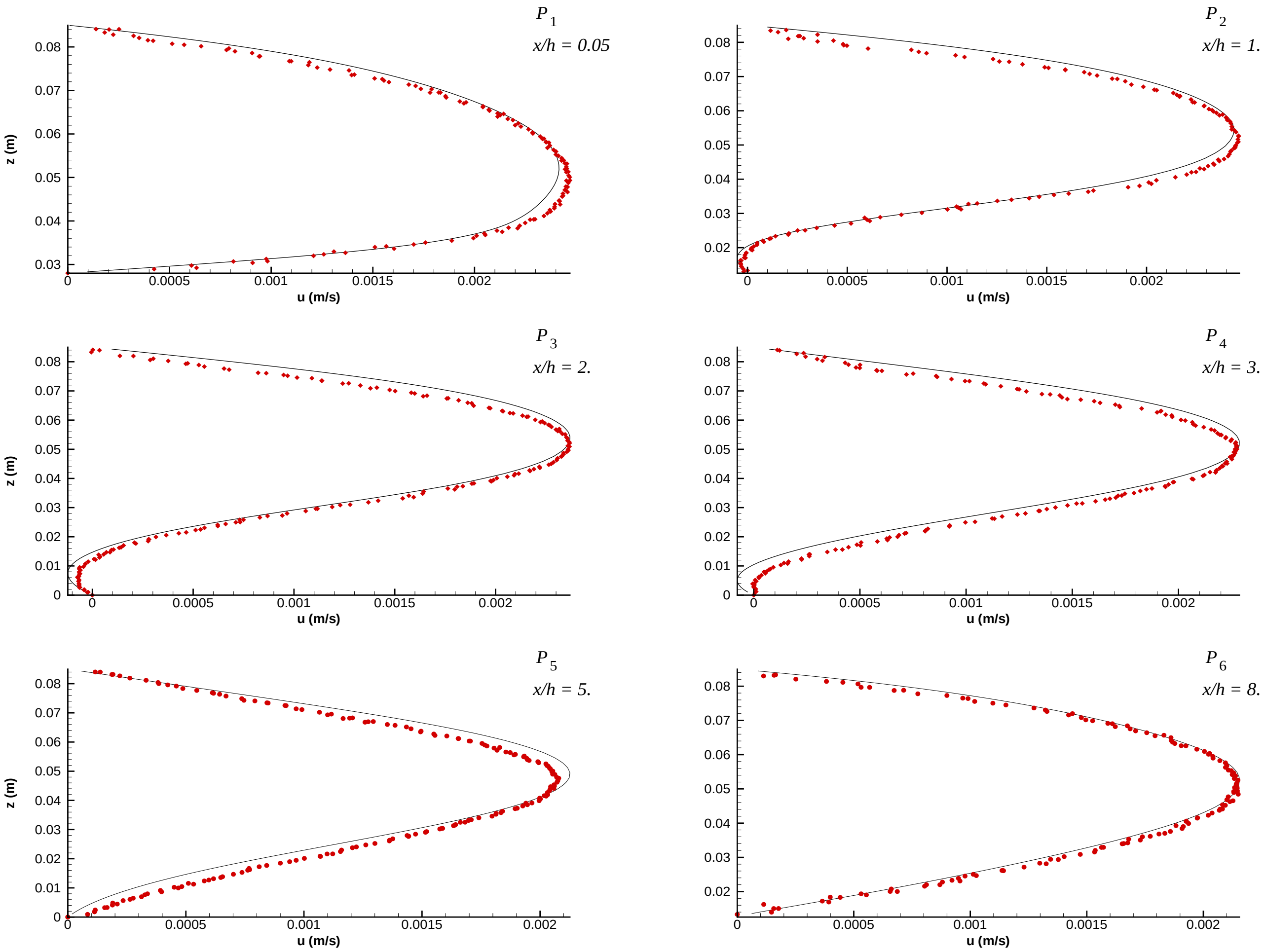

Figure 4.15: Axial velocity profiles with Saturne (——) and Spartacus-2D (•) for the hill flow.

vergence: the initial particle distribution is considered as a fixed grid and the instantaneous results are interpolated on the grid, as done in classical Eulerian codes. The average velocity fields are much closer to those obtained with Saturne, as shown in figure 4.16. Figure 4.17 also reveals a smoother recirculation zone. The axial velocity profiles of figure 4.15 are much smoother and closer to the ones of Saturne when the temporal average is considered. Indeed, figure 4.18 shows a close agreement between the two codes. The reattachment point is now predicted with an error of 2.1 %.

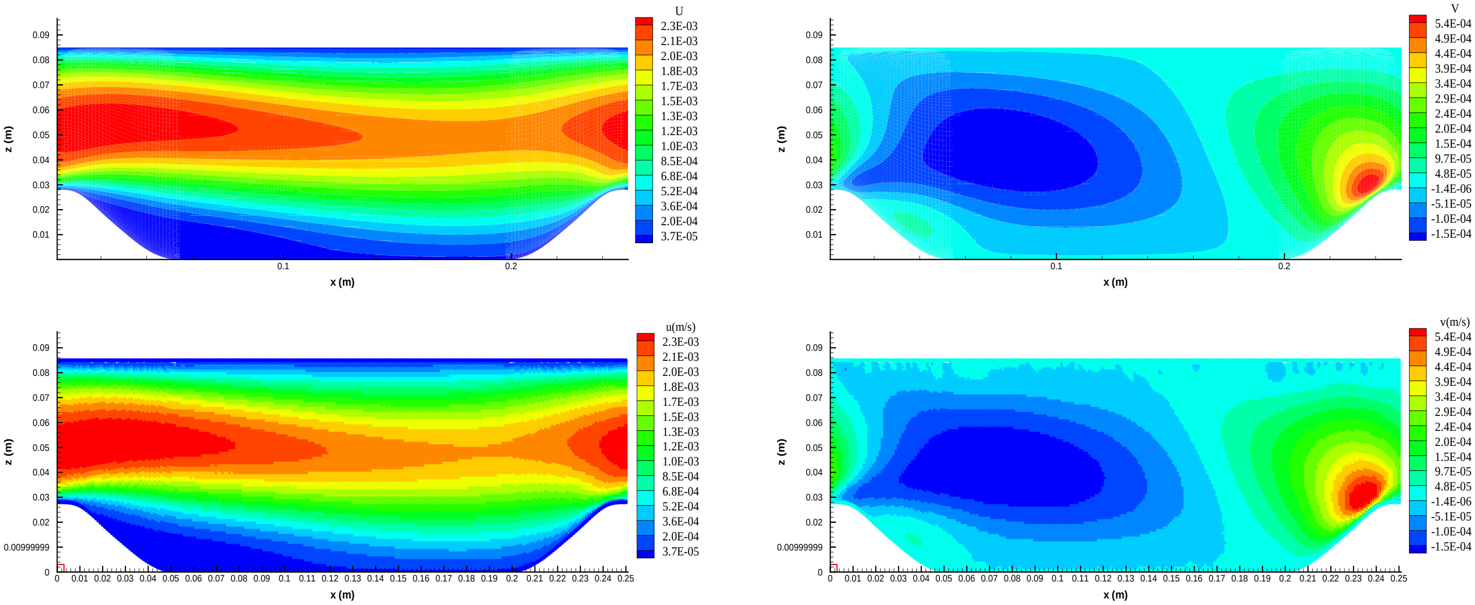

Figure 4.16: Time averaged axial (left) and vertical (right) velocity fields with Saturne (top) and Spartacus-2D (bottom) for the hill flow.

4.3.4 Partial conclusion

It has been shown in this part that the simulation of a laminar separated flow by Spartacus-2D is satisfactory. However, in order to smooth the instabilities due to the Lagrangian characteristic of SPH, a temporal average should be used. In this case, the velocity profiles are very close to those of an Eulerian code and the characteristics of the separation zone are well predicted. However, the slightly shorter recirculation and less intense backflow seem to correspond to a somewhat “more viscous” case: it is possible that the randomness of the Lagrangian method acts as additional Brownian motion and subsequent mixing effect. This will be highlighted further in the LES section. Finally, the numerical diffusion is likely to be larger since SPH uses a constant mesh step whereas Saturne uses near wall refinement. In this context, the 2 % error of the recirculation length can be considered quite satisfactory.

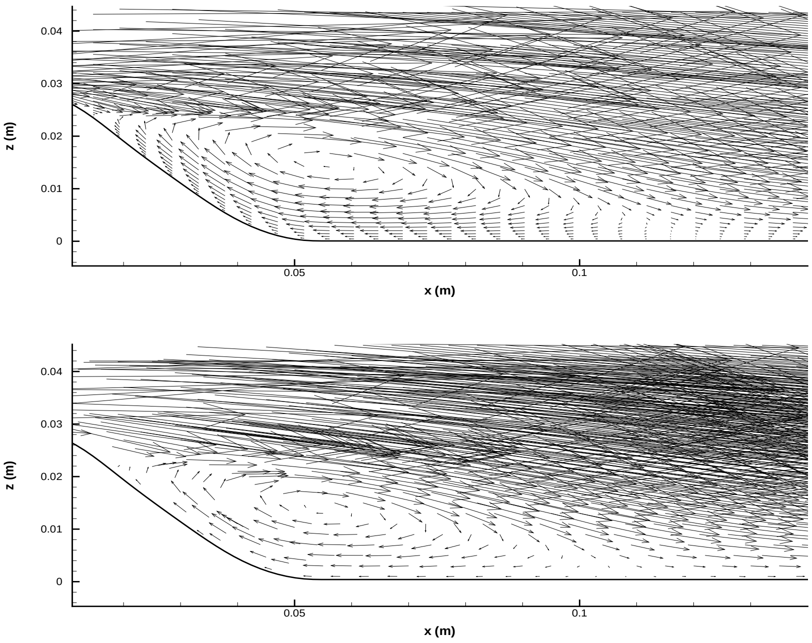

Figure 4.17: Time averaged velocity vectors in the vicinity of the downhill side with Saturne (top) and Spartacus-2D (bottom) for the hill flow.

4.4 Laminar flow in a 2D backward-facing step geometry[4]

The aim of this part is to reveal the ability of Spartacus-2D to deal with strongly separated flows. This fourth test case therefore corresponds to a laminar flow in a classical 2D backward facing-step geometry.

4.4.1 Geometry of the system

The geometry of the backward-facing step is based on the one used by Armaly [2]. However, for reasons of convenience, another step is added at the end of the channel in order to apply periodic conditions with respect to x-direction. The geometry is thus characterised by three bounded channels: an "inlet", "central" and "outlet" channel, as described in figure 4.19. Contrary to the previous 2D hill channel, the lengths l and L must be large enough: in the "inlet" channel, the axial velocity field must be fully parabolic and the "outer" channel must not disturb the flow in the vicinity of the backward-facing step. The dimensions of the system are defined in table 4.12.

[4] Presented at "Institute for Computational Fluid Dynamics" conference (Oxford, 2004) and to be published in a special issue of "International Journal for Numerical Methods in Fluids".

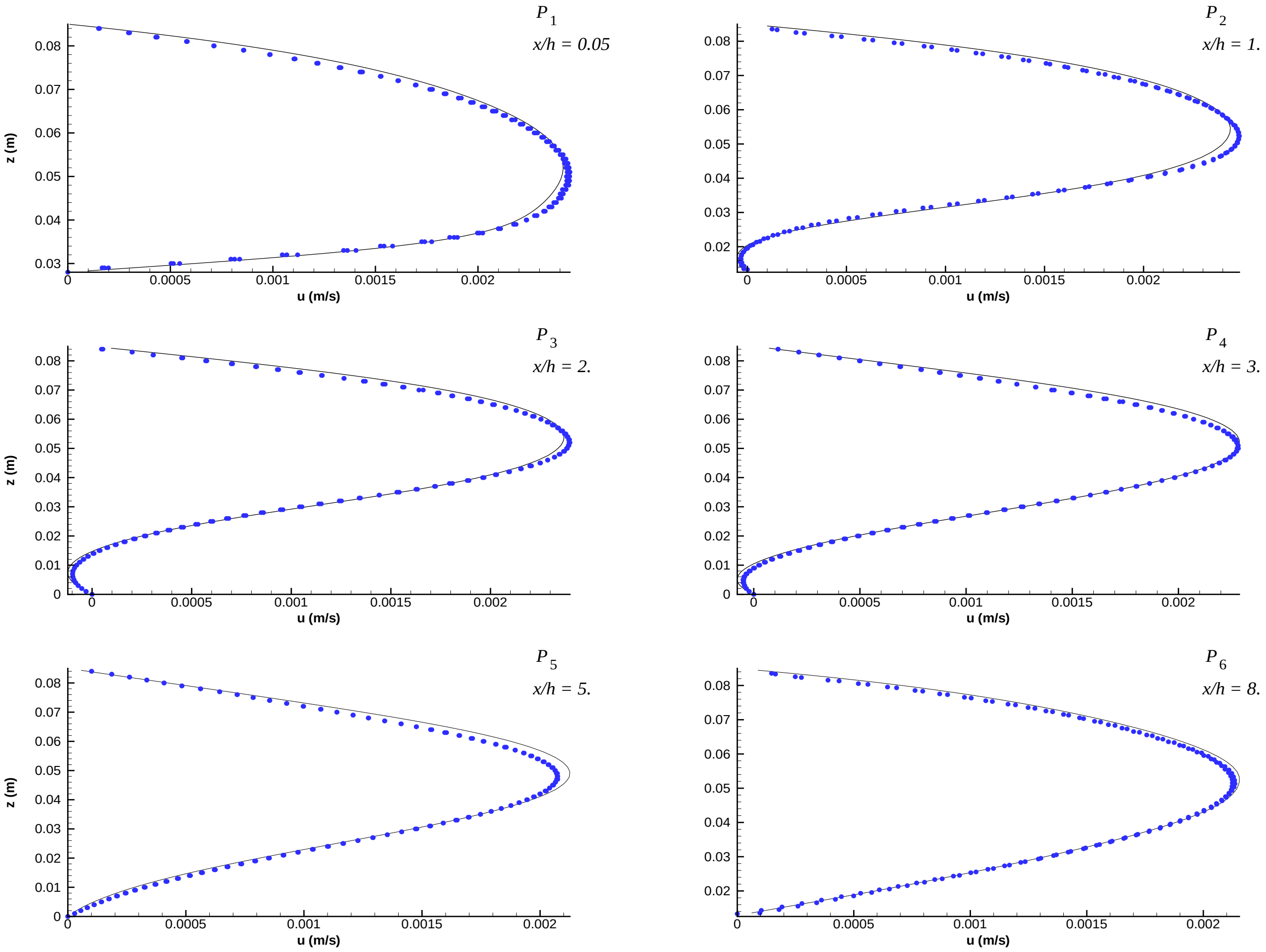

Figure 4.18: Time averaged axial velocity profiles obtained with Saturne (——) and Spartacus-2D (●) for the hill flow.

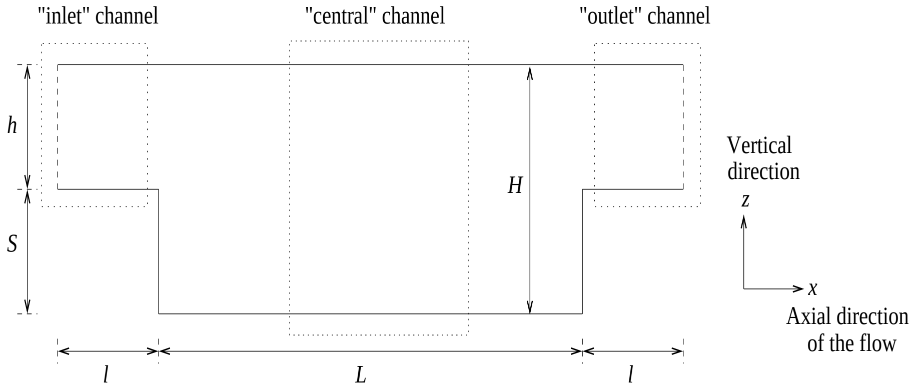

Figure 4.19: Geometry of the 2D backward-facing step.

h (mm)	5.2
S (mm)	4.9
l (mm)	16.3
L (mm)	89.8
Expansion ratio: $(h+S)/S$	1.94

Table 4.12: Dimensions of the 2D backward-facing step.

Reynolds number : Re	100
Mean bulk velocity : U $(m{\cdot}s^{-1})$	0.14

Table 4.13: Physical characteristics of the backward-facing step flow.

4.4.2 System modelling

Simulation conditions

The physical characteristics of the backward-facing step flow are presented in table 4.13, where the Reynolds number is defined according to Armaly [2] by

$$Re = \frac{UD}{\nu} \tag{4.27}$$

U denotes the mean bulk velocity in the "inlet" channel, D is twice the "inlet" channel height $(= 2h)$ and ν corresponds to the kinematic viscosity of the fluid.

Fluid discretisation and wall modelling

The fluid discretisation is described in table 4.14. The particle initial spacing δr is decided from that 30 particles are placed along the step height. The wall modelling is identical to the previous case, with wall particles and three layers of mirror particles. Once again, no particle refinement is achieved near the wall, all the particles are regularly distributed on a cartesian grid. The initial and boundary conditions are identical to those of the previous hill flow simulation.

Fluid particles	39 751
Wall particles	1 526
Mirror particles	4 686
Total particle number	45 999
Particle initial spacing: δr (mm)	49/300

Table 4.14: Fluid, wall and mirror particle discretisation for the backward-facing step flow.

Kernel order	4
$h/\delta r$	1.2
Speed of sound : c_0 $(m{\cdot}s^{-1})$	2.

Table 4.15: Main SPH numerical parameters for the backward-facing step flow.

SPH equations and numerical parameters

The SPH equations are also identical to those used for the hill flow test case and the fluid is once again driven by an external horizontal force in order to impose the correct mass flow rate. The fluid is still modelled by a nearly incompressible one, which implies the use of the state equation (3.17) to determine the particle pressure. The main SPH numerical parameters are presented in table 4.15: the smoothing length value has been determined through numerical tests [46] and the speed of sound is equal to ten times the maximal velocity of the flow.

4.4.3 Simulation results

Since there is no theoretical solution for this problem, the results obtained with Spartacus-2D are compared to those obtained with the Eulerian code *Fluent*, based on the finite volume method. Before the establishment of a steady state, large voids appear next to the step, as shown on the left picture of figure 4.20. This unphysical phenomenon could be a consequence of a tensile instability as described in part 2.8.1: indeed, the voids of figure 2.19 which are due to tensile instabilities are very close to those exhibited by figure 4.20. Since the number of particles is constant throughout the simulation, the existence of voids reveals that some particles are too close to each other in other zones. This particle clumping means therefore that the pressure gradient term, which acts as a repulsive force between particles, is not sufficient to prevent the clumping, characteristic of the tensile instability. This first result shows a limitation of the nearly incompressible assumption: the pressure values are not well enough computed to maintain the cohesion of an incompressible flow [36], [46]. Since the pressure gradient discretised according to equation (3.2) involves a summation of the pressure of the particles, an alternative to avoid the particle clumping is to increase the pressure of all the particles by considering a larger speed of sound. As mentionned in part 2.4.3, the pressure will then respond in a stronger way to variation in density, which is consistent with the work described in [64]. In this case, a speed of sound forty times (instead of ten) higher than the maximal velocity of the flow does not induce artificial void (see the right picture of figure 4.20). To confirm the validity of the results for the higher speed of sound, four velocity profiles obtained with Spartacus-2D are compared with the results of Fluent. These profiles are represented in figure 4.21 and located in table 4.16. Firstly, figure 4.22 proves that the flow in the "inlet" channel is parabolic and fully developed: the distance l is thus long enough. Figure 4.23 reveals that the axial velocity field in the "central" channel is also fully parabolic.

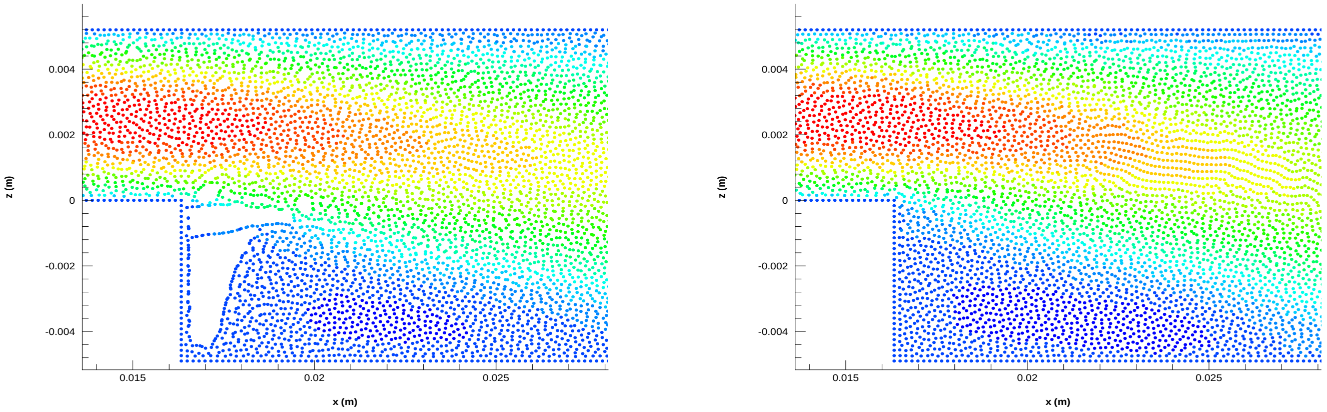

Figure 4.20: Artificial voids due a tensile instability (left) and correction with a higher speed of sound (right) for the backward-facing step flow.

Therefore, there is no influence of the "outlet" channel on the recirculation zone: the distance L is long enough. Next to the step, the recirculation zone is quite clear, as one can notice from the left picture of figure 4.24. Besides, the axial velocity profiles of this zone are quite satisfactory as well (see the right picture of figure 4.24). The dimensionless analysis of the reattachment point prediction X_R is shown in table 4.17. Spartacus-2D has an error of 4.8 % compared to Fluent. Once again, a temporal average seems to be necessary to smooth the Lagrangian instabilities. Indeed, figures 4.25, 4.26, 4.27 reveal a better agreement with the two codes when a temporal average is achieved. The error on the reattachment point is equal this time to 2.6 %. The profile across the free shear layer (figure 4.27) tends to show a less sharp velocity gradient with Spartacus-2D, which again makes one think of additional mixing effects due to the unsteadiness of the scheme. As adverse pressure gradients are much stronger than in the previous hill case, compressibility effects may also result in differences with the incompressible Fluent simulation. Indeed, figures 4.23 and 4.26 show an underestimation of the flow rate across the centre of the domain whereas the inlet flow rate (see figures 4.22 and

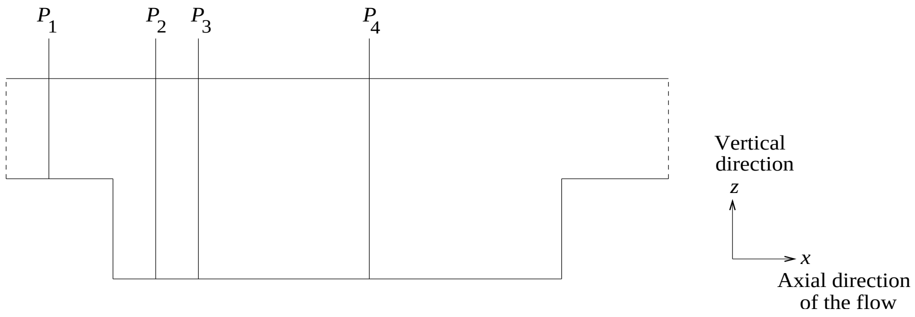

Figure 4.21: Position of the four considered profiles for the backward-facing step flow.

Profiles	x/S
P_1	1.0
P_2	3.7
P_3	4.1
P_4	10.2

Table 4.16: Position of the four considered profiles for the backward-facing step flow.

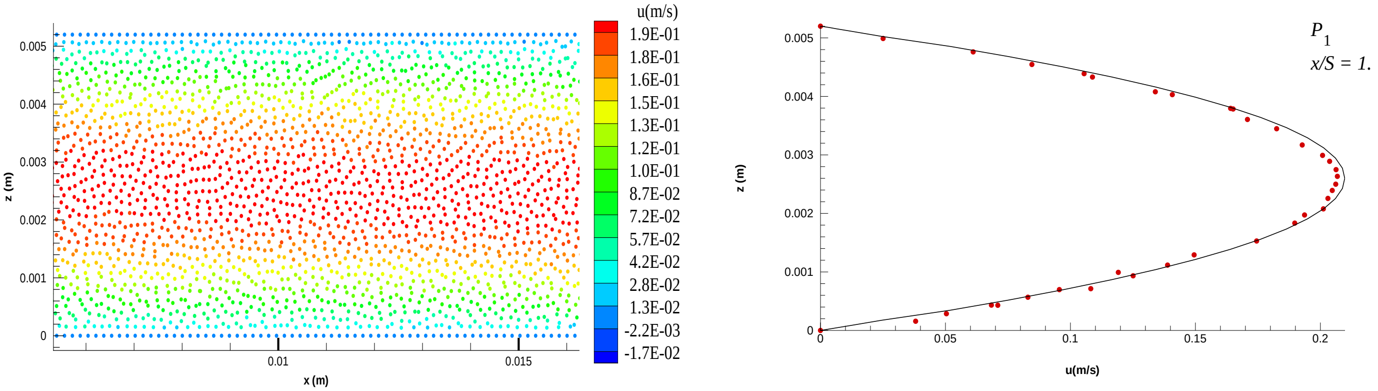

Figure 4.22: Axial velocity field in the "inlet" channel (left) and axial velocity profile relative to P_1 (right) with Fluent (——) and Spartacus-2D (•).

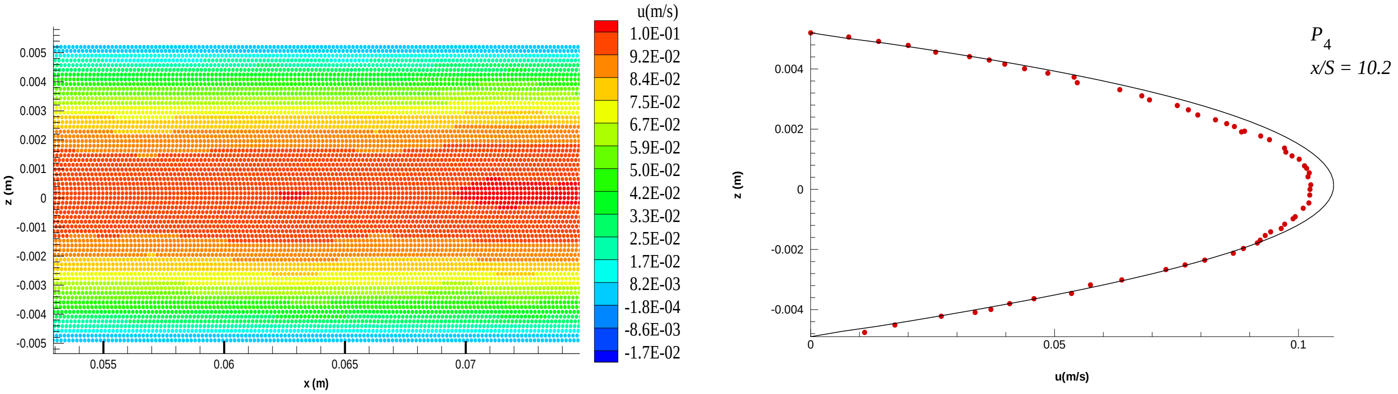

Figure 4.23: Axial velocity field in the "central" channel (left) and axial velocity profile relative to P_4 (right) with Fluent (——) and Spartacus-2D (•).

	Fluent	Spartacus-2D	Relative difference (%)
Ratio X_R/S	6.3	6.0	4.8

Table 4.17: Reattachment point prediction for the backward-facing step flow.

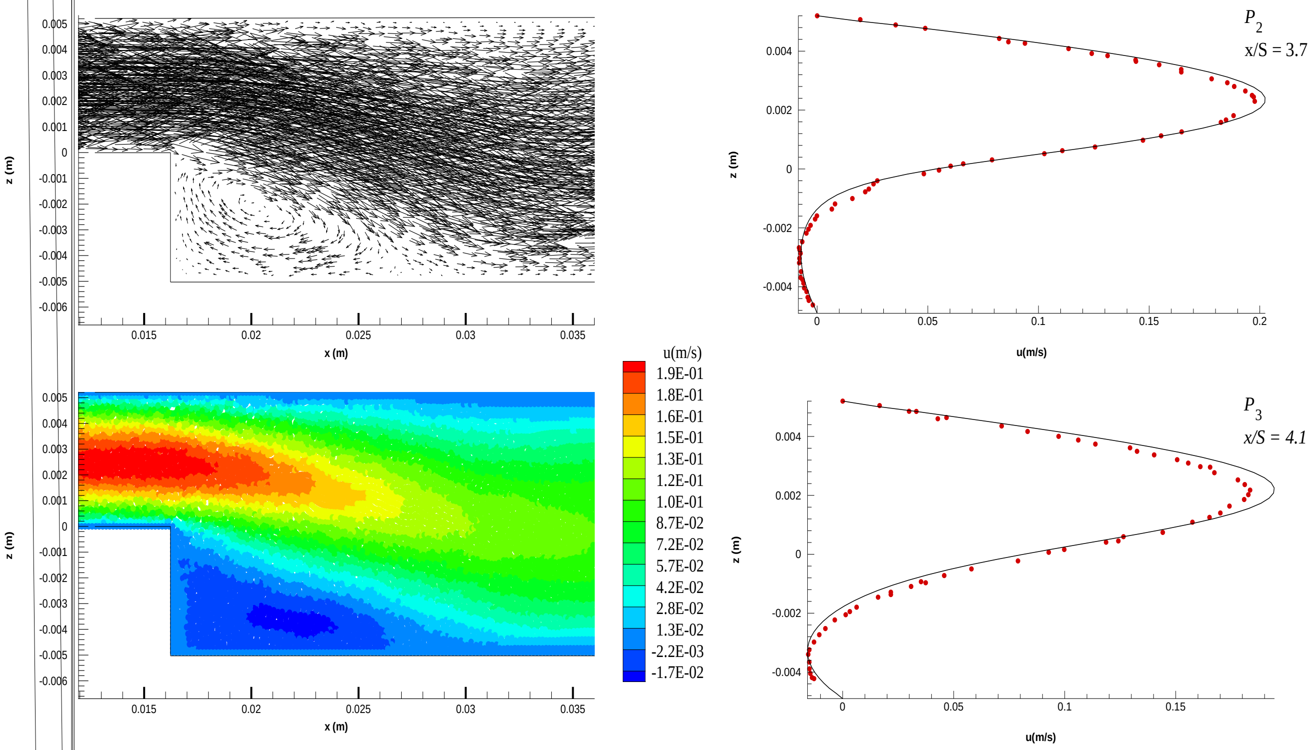

Figure 4.24: Velocity field (left) next to the step and axial velocity profile relative to P_2, P_3 (right) with Fluent (——) and Spartacus-2D (●).

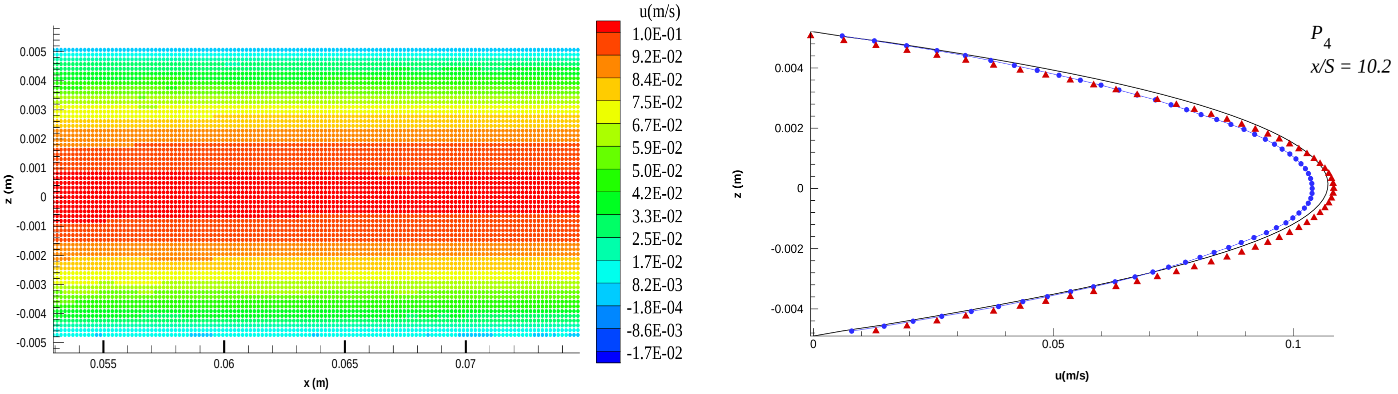

Figure 4.26: Time averaged axial velocity field in the "central" channel (left) and time averaged axial velocity profile relative to P_4 (right) with Fluent (——), Spartacus-2D (●) and theoretical values (Δ).

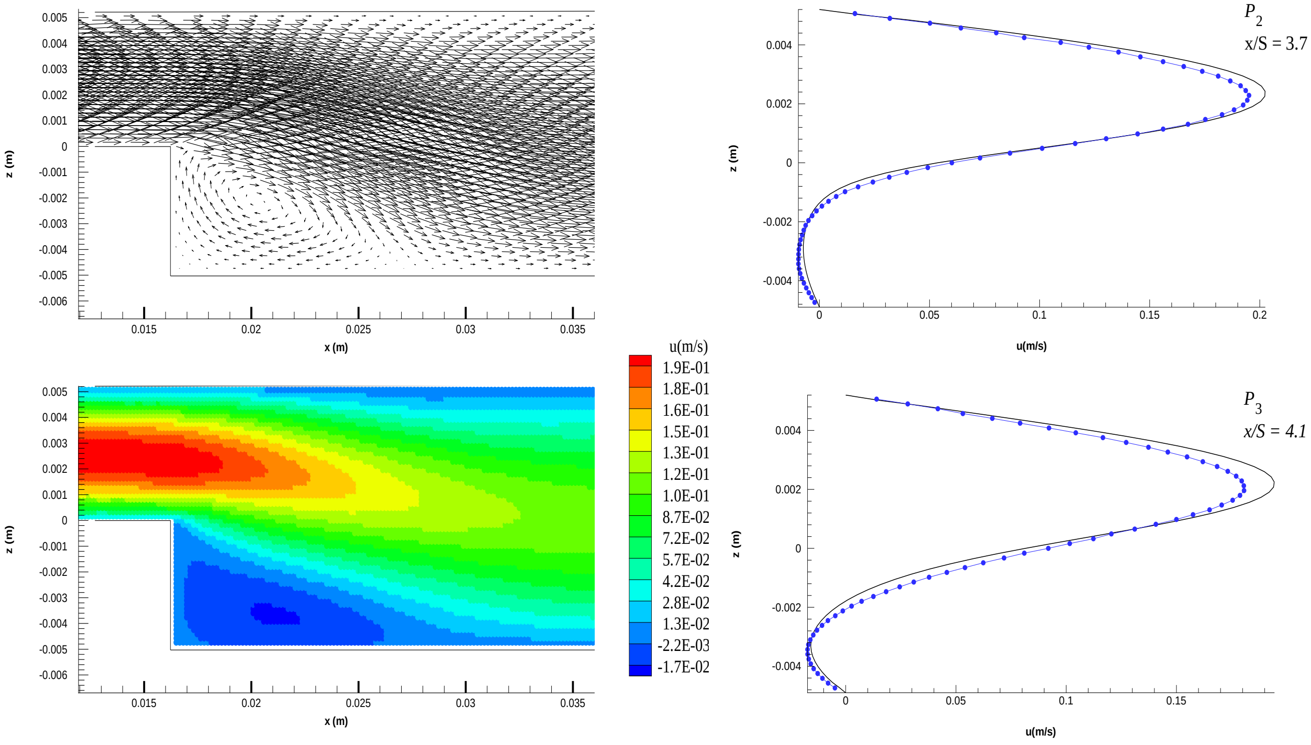

Figure 4.27: Time averaged velocity field (left) next to the step and time averaged axial velocity profile relative to P_2, P_3 (right) with Fluent (——) and Spartacus-2D (●).

4.25) seems correct.

4.4.4 Partial conclusion

A limitation of the nearly incompressible assumption has been revealed through this test case: the pressure field is not perfectly computed, inducing tensile instabilities. It should be mentioned that numerous simulations were performed on this test case (by Eun-Sug Lee, a MSc student of UMIST [46]) in an attempt to prevent the creation of voids, for instance adding hydrostatic pressure, or testing various kernels, time steps, and wall treatments. In the end, only increasing the speed of sound (at high computational costs) could resolve the problem. Then, the axial velocity profiles and the reattachment point prediction are satisfactory. Spartacus-2D is so able to deal with sharp geometry and once again, no wall refinement is required.

Chapter 5

Turbulence modelling through SPH

This chapter firstly introduces some basic notions of turbulent flows: the main characteristics of these flows and some energy aspects are presented. Turbulence modelling in CFD on one hand and through SPH on the other hand is also discussed.

5.1 General introduction to turbulence

5.1.1 Turbulent flows in everyday life, environment and industry

There are many opportunities to observe turbulent flows in our everyday surroundings: the smoke of a cigarette, the flow inside a cup of coffee when stirring are turbulent flows. Nature also gives us an intuitive knowledge of turbulence in fluids: the flow of a waterfall (see the left picture of figure 5.1), the wave impacting a shoreline (see the right picture of figure 5.1), the fluid behaviour in a river (see the left picture of figure 5.2) correspond to turbulent motions. At larger scales, the cyclones (see the right picture of figure 5.2) are turbulent flows as well. In engineering applications, they are also prevalent: in the processing of liquids or gases with pumps, compressors, pipelines. Similarly, the flows around vehicles, e.g. airplanes, automobiles (see figure 5.3), ships, submarines are turbulent.

5.1.2 Characteristics of a turbulent flow

Unpredictability

Firstly, a turbulent flow is disordered in time and space: when observing a waterfall, one immediately sees that the flow is unsteady, seemingly random and that the motion of every droplet is unpredictable [69]. Experimentally, Tong [80] also revealed that the axial component of the velocity $U_1(t)$ on the centerline of a turbulent jet looks like a random signal (see figure 5.4). A more mathematical definition of the unpredictability of a turbulent flow is given by Lesieur in [47]. If a system is deterministic in the Lagrange sense, i.e. if its initial velocities are known everywhere at a given time t_0, then there is only one solution for any time $t > t_0$

Figure 5.1: “Grand-Bassin” water fall (left) and wave impacting the shoreline of “Saint-Leu” in Réunion Island (right).

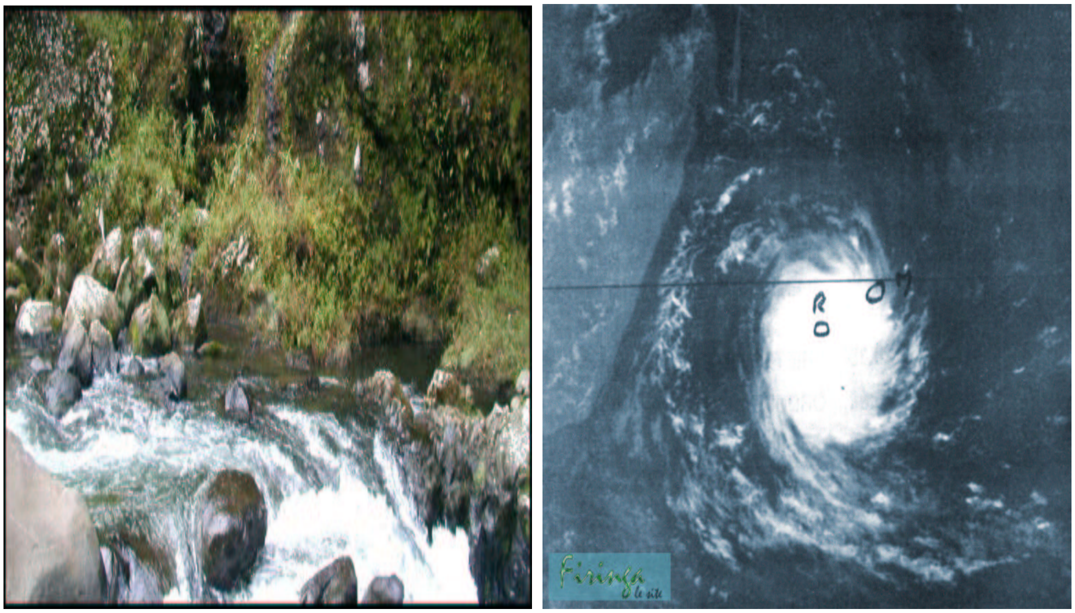

Figure 5.2: Turbulent flow in the river “Bassin la Mer” at “Bras-Panon” (left) and satellite picture of the “Firinga” cyclone (right) near Réunion Island.

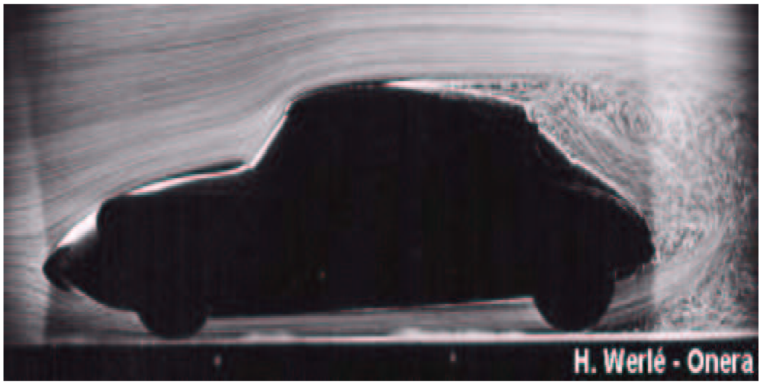

Figure 5.3: Turbulent flow around a car (Onera picture).

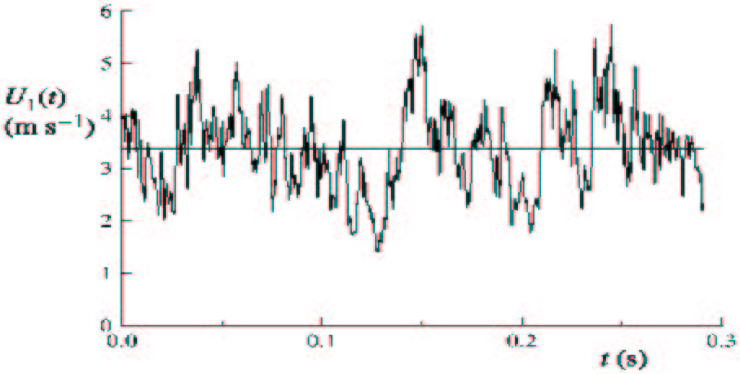

Figure 5.4: Time history of the axial component of velocity $U_1(t)$ on the centerline of a turbulent jet [80].

(existence and uniqueness theorem in Mathematics). Considering two initial possible states very close to each other, the system will be unpredictable when the difference between them will grow significantly in time, to reach finite values. An accurate deterministic prediction of the system evolution is thus impossible. This is nothing more than the concept of deterministic chaos studied in dynamical systems [88]. An exemple of such unpredictability is the famous *butterfly effect*, introduced by the meteorologist E. Lorentz: the motion of a butterfly wing in Australia would eventually change the climate in the northern hemisphere.

Mixing and transport properties

The second important characteristic of turbulence is its ability to mix and transport quantities much more rapidly and efficiently than laminar flows do:

- The effectiveness of turbulence for mixing fluids is of prime importance in many applications: the mixing of fuel and air in engines, the mixing of the reactants in chemical reactors (see figure 5.5), pollutant streams mixing into the atmosphere, are much more effective when turbulent effects are added to molecular diffusion processes [69]. This is well demonstrated by the famous experiment of Osborne Reynolds (1883) [70].

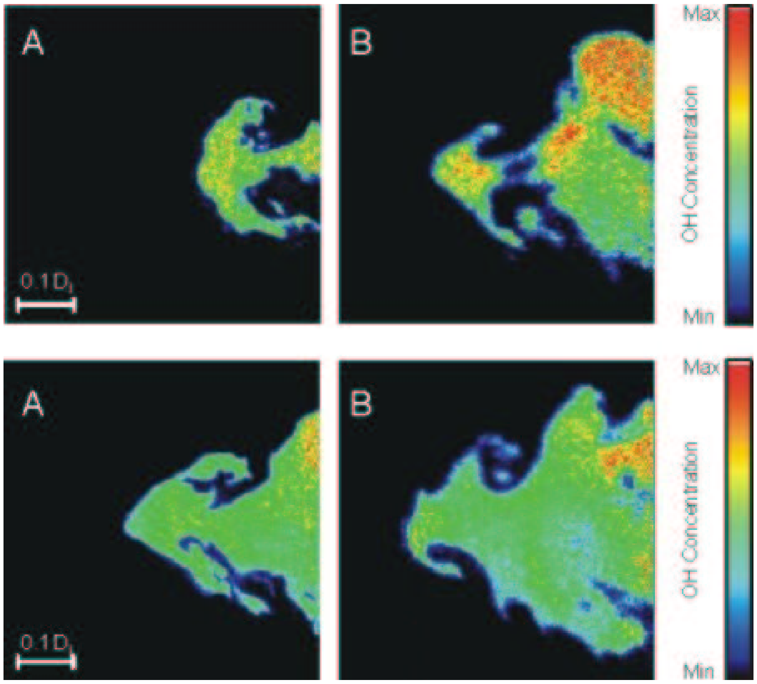

Figure 5.5: OH concentration in a combustion chamber visualised by Laser Induced Fluorescence.

- Turbulence is also effective at transporting the momentum of the fluid. As a consequence, on aircraft's wings and ships' hulls the wall shear stress (and hence the drag) is much larger than it would be if the flow were laminar.

- Similarly, rates of heat transfer at solid-fluid interfaces are much enhanced in turbulent flows [34], [83].

Wide range of eddies

In a turbulent flow, motions of many scales are observed: considering the waterfall of figure 5.1, large eddies comparable in size to the waterfall width on one hand, and smaller scales which can not be resolved by the camera on the other hand, are present in the flow. This third characteristic is due to the non-linear term of the momentum equation (5.1) [54].

$$\frac{\partial \underline{u}}{\partial t} + \underbrace{\underline{u}.\nabla \underline{u}}_{non-linear} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \underline{u} \tag{5.1}$$

In this equation, p denotes the pressure, $\underline{u}$ the velocity field and ν the kinematic viscosity of the fluid.
The major motivation for the study of turbulent flows is the second property: the enhancement of transport and mixing of species, momentum, heat transfer due to turbulent effects. Indeed, the engineer, for instance, is mainly concerned with the knowledge of turbulent heat exchange coefficients, or the turbulent drag.

5.1.3 Energy aspects of turbulence

The Kolmogorov theory

A turbulent flow contains many coherent structures, as shown on figure 5.6: they are often referred to as eddies, since they are usually associated with rotating motions of the flow. One fundamental result of turbulence [47], [69], [79], [95] is that these eddies are not all of one particular size, but that a broad range of large to small eddies exist, as one can notice from figure 5.6. Moreover, these structures are also characterised by different time scales. These wide ranges of length and time scales for turbulent flows are due to the non-linear term of the momentum equation (5.1).
In 1941, Kolmogorov intuitively introduced three assumptions in order to investigate turbulent flows. They are reported by Pope [69] and are usually simplified according to:

1. The major part of the turbulent kinetic energy k is transported by the large scales, which are not influenced by the viscosity.

2. The viscosity only affects the small scales, which are responsible for the major part of the dissipation.

3. The energy dissipated by the small scales comes from the large ones.

These three assumptions are at the heart of Kolmogorov's theory and enable us to quantify the length and time scales of the turbulent structures.

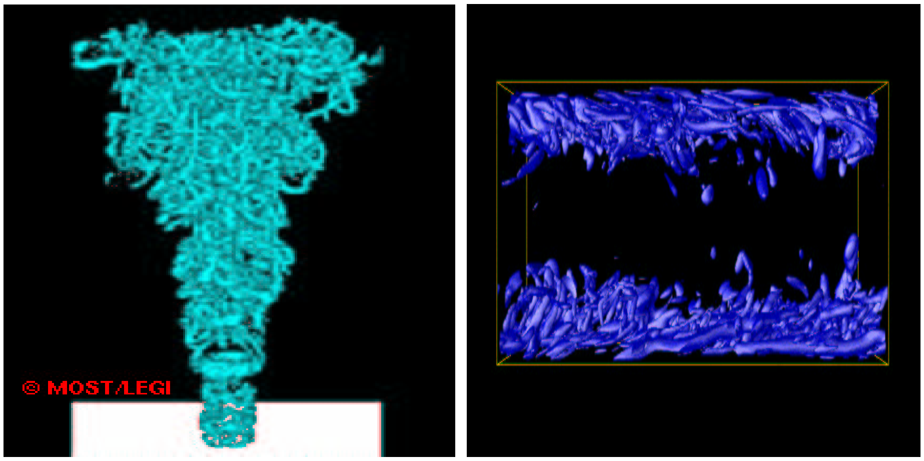

Figure 5.6: Turbulent coherent structures in a jet (left) [75] and in a channel (right) performed by A. Tanière.

The energy cascade

Contrary to the large scales which extract their turbulent kinetic energy directly from the mean flow [54], the smaller structures are fed by a continuous decay of the unstable large eddies: the large scales break up into smaller ones with a negligible amount of energy loss (provided the Reynolds number is high enough [69]). This process, due once again to the non-linear advection term, has been introduced in 1922 by Richardson as the “energy cascade”. This structure breaking up continues until the smallest scales of the flow, which finally dissipate the energy of the turbulent motion through the effect of viscosity. In other words, the energy cascade continues until the Reynolds number $Re_l = ul/\nu$ (based on a characteristic length and velocity scales of the small eddies) is sufficiently low that the eddy motion is stable and molecular viscosity is effective in dissipating the kinetic energy. The dissipation is thus placed at the end of a sequence of processes. The reason for this is that only at small length scales, the turbulent velocity gradients are large enough to result in any significant effect of dissipation. The process of the energy cascade is illustrated in figure 5.7.

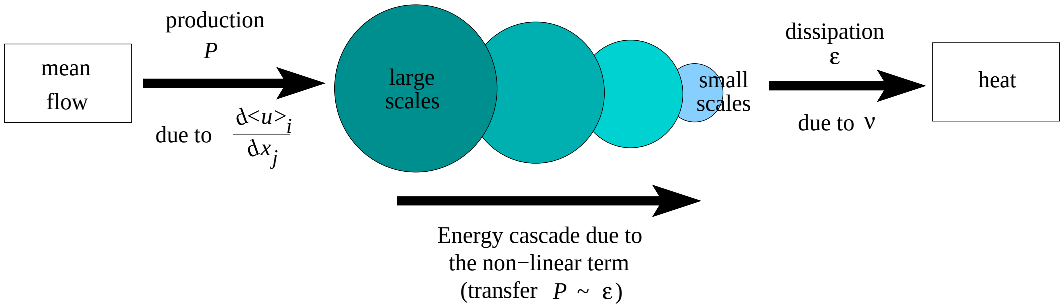
mean flow
production
P
due to $\frac{\partial \langle u \rangle_i}{\partial x_j}$
large scales
small scales
dissipation
ε
due to ν
heat
Energy cascade due to the non−linear term (transfer $P \sim \varepsilon$)

Figure 5.7: Cascade energy scheme [88].

Scales of large eddies

The size L_i of the largest eddies, called the integral size, is determined by the geometry of the flow configuration: typical values of L_i for wall-bounded shear-driven turbulence are $L_i \sim 0.1L$, where L denotes a characteristic length scale of the system geometry (e.g. the pipe diameter or the channel height). These large eddies are also characterised by a velocity scale u_t. The Reynolds number for the large eddies $Re_{L_i} = u_t L_i/\nu$ is large so the direct effects of viscosity are negligibly small (according to the first assumption of Kolmogorov's theory). The loss of kinetic energy of the large scales is characterised by the dissipation rate ϵ (per unit mass): it is independent of the small scale and the fluid properties since it is fully determined by the large scale characteristics. This is expressed by the following relation which is a fundamental result in turbulence theory

$$\epsilon \approx \frac{u_t^3}{L_i} \tag{5.2}$$

This relation, consistent with the experimental observations in free shear flows [69], can be interpreted as the ratio of the kinetic energy of the large eddies ($\approx u_t^2$) and their life-time (L_i/u_t).

Scales of small eddies

The length and velocity scales of the smallest eddies are determined by the amount of kinetic energy transferred along the energy cascade (or equivalently by the dissipation rate ϵ) and the kinematic viscosity ν of the fluid. As a result of dimensional analysis, one obtains the following expressions for the length scale η_k, the velocity scale u_k and the time scale τ_k of the smallest eddies, known as the Kolmogorov scales

$$\eta_k = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}} \tag{5.3}$$

$$u_k = (\nu\epsilon)^{\frac{1}{4}} \tag{5.4}$$

$$\tau_k = \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}} \tag{5.5}$$

From the previous three relations, it follows that the Kolmogorov Reynolds number $Re_k = \eta_k u_k/\nu$ is unity. This is consistent with the notion that the cascade proceeds to smaller and smaller scales until the Reynolds number defined on small scale characteristics is small enough for dissipation to be effective.

Relations between large and small scales

Since the dissipation rate is known in terms of the large scale properties, one can easily deduce relations between the various scales of the large and small eddies: combining equations (5.2), (5.3), (5.4) and (5.5) yields

$$\frac{\eta_k}{L_i} \approx Re_{L_i}^{-\frac{3}{4}} \tag{5.6}$$

$$\frac{u_k}{u_t} \approx Re_{L_i}^{-\frac{1}{4}} \tag{5.7}$$

$$\frac{\tau_k}{L_i/u_t} \approx Re_{L_i}^{-\frac{1}{2}} \tag{5.8}$$

with

$$Re_{L_i} \approx \frac{u_t L_i}{\nu} \tag{5.9}$$

When Re_{L_i} increases, the small scales become much smaller than the large ones.
The appearance of a broad range of scales in a turbulent flow is reflected in the principles of numerical simulation techniques used to investigate turbulent flows.

5.2 Turbulence simulation and modelling

5.2.1 Direct Numerical Simulation

Principle

The first approach for turbulence simulation is to solve the Navier-Stokes equations without any averaging or approximation [20]. By resolving all the scales of motion, the result is equivalent to a short-duration laboratory experiment. This approach is called Direct Numerical Simulation (DNS) and when it can be applied, it is unrivalled in accuracy and in the level of description provided.

Applications

DNS studies have proved extremely valuable in supplementing our knowledge from experiments of turbulence [69]. For instance, the details of near-wall flows and coherent structures are more easily studied with DNS than with experiments (see figure 5.8). It is also a very useful tool that can give a better insight for turbulence modelling. For instance, the numerical constants present in some statistical models can be determined thanks to DNS (see part 5.2.2).

Computational limitation

When achieving a DNS, the calculation domain must be at least a few times the integral scale L_i. Moreover, in order to capture the small structures responsible of the viscous dissipation, the grid must be no larger than the Kolmogorov scale η_k. For homogeneous turbulence, the simplest type of turbulence, there is no reason to use anything other than a uniform grid. In this case, the number of grid points in each direction must be at least L_i/η_k, which is equal to $Re_{L_i}^{3/4}$ according to equation (5.6). Since this number of points must be employed in each of the three coordinate directions, the total number of grid points of such simulation scales as $Re_{L_i}^{9/4}$. This means that DNS can be carried out only at relatively low Reynolds numbers ($Re_{L_i} \approx 100$), which allows one to reach the low end of the range of Reynolds numbers of engineering interest. Moreover, as mentioned before, a turbulent flow presents a wide range of time scales: the time step used for a DNS must then be fitted to the smallest time scale of the problem, i.e. the Kolmogorov time scale τ_k. It has been shown by Nieuwstadt [67] that the required computer memory M_c (in bytes) and the computer time T_c (in CPU-seconds) evolve according to

$$M_c = 10^5 Re_{L_i}^{\frac{9}{4}} \tag{5.10}$$

$$T_c = 0.3 Re_{L_i}^{3} \tag{5.11}$$

These relations reveal that DNS is obviously limited by the computer capacity.

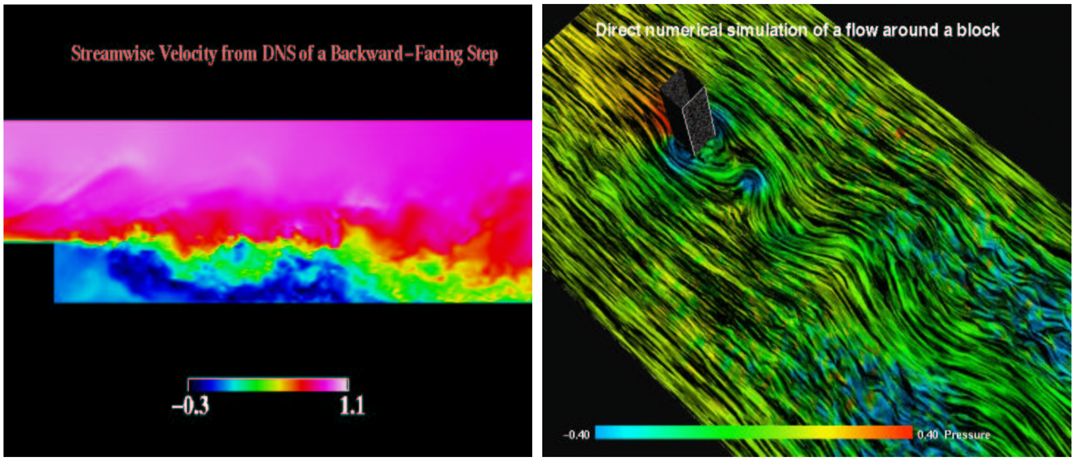
Streamwise Velocity from DNS of a Backward-Facing Step
−0.3
1.1
Direct numerical simulation of a flow around a block
-0.40
0.40 Pressure

Figure 5.8: Streamwise velocity over a backward-facing step (left) by H. Le and P. Moin (Stanford University) and vorticity field in wake of a square cylinder (right) by R. Verstappen (University of Kroningen), with DNS.

5.2.2 Reynolds-averaged modelling

The second way to investigate turbulent flows is to time-average unsteady fluctuations, hence achieving turbulence statistical modelling. The simplest statistical point of view consists of considering only averaged values, which are defined by ensemble, time or spatial averaging [20], [54]. By applying the average operator $\langle\ \rangle$ to the Navier-Stokes equations, one ends up with the Reynolds equations (1894) [69], [95]. The turbulence statistical models based on the Reynolds equations are thus called RANS models (Reynolds Averaged Navier-Stokes equations).

The incompressible Reynolds equations

In an incompressible Reynolds formalism, the Reynolds equations are written

$$\frac{\partial\langle u_i\rangle}{\partial x_i} = 0 \tag{5.12}$$

and

$$\frac{d\left\langle u_i\right\rangle}{dt} = \frac{\partial\left\langle u_i\right\rangle}{\partial t} + \left\langle u_j\right\rangle\frac{\partial\left\langle u_i\right\rangle}{\partial x_j} = -\frac{1}{\rho}\frac{\partial\left\langle p\right\rangle}{\partial x_i} + \nu\frac{\partial^2\left\langle u_i\right\rangle}{\partial x_j\partial x_j} - \frac{\partial R_{ij}}{\partial x_j} + F_i^e \tag{5.13}$$

with a summation over j subscript (Einstein's conventions), where i varies from 1 to 3 in 3D and from 1 to 2 in 2D. $\langle . \rangle$ denotes average values and R_{ij} are the components of the Reynolds stress tensor defined by $R_{ij} = \left\langle u_i' u_j' \right\rangle$. The velocity fluctuations are denoted $u_i' \equiv u_i - \langle u_i\rangle$ and must satisfy $\langle u_i'\rangle = 0$. F_i^e corresponds to the i-component of an external volumetric force. The Reynolds equations (5.12) and (5.13) describe the evolution of the mean flow quantities. The Reynolds stress tensor incorporates the influence of the removed turbulence fluctuations on the mean flow and describes the influence of all scales of turbulent motion, including the anisotropic large scales [17].

The eddy viscosity assumption

In order to close the equation set relative to (5.12) and (5.13), several techniques allow to express the Reynolds stress values in term of the resolved quantities: the more commonly used one is based on the eddy viscosity assumption, introduced by Boussinesq in 1880. In laminar flows, energy dissipation, transport of mass and momentum are all mediated by the viscosity. Since these phenomena are enhanced by turbulence (see part 5.1.2), it is thus natural to assume that turbulent effects can be represented by an increased viscosity ν_T, which models the diffusion and dissipation properties of eddies. The average field stability is then ensured by this eddy viscosity ν_T, generally much higher than the fluid molecular viscosity. With this eddy viscosity assumption, the Reynolds stresses are modelled by

$$R_{ij} \equiv \left\langle u_i^{'} u_j^{'} \right\rangle = \frac{2}{3} k \delta_{ij} - 2\nu_T \langle s_{ij} \rangle \tag{5.14}$$

where $k = \left\langle u_i^{'} u_i^{'} \right\rangle / 2$ denotes the turbulent kinetic energy (per mass unit), δ_{ij} is the Kronecker's symbol and $\langle s_{ij} \rangle$ the components of the rate of strain tensor based on averaged velocities

$$\langle s_{ij} \rangle = \frac{1}{2} \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) \tag{5.15}$$

Then, equation (5.13) becomes

$$\frac{d \langle u_i \rangle}{dt} = -\frac{1}{\rho} \frac{\partial}{\partial x_i} \left(\langle p \rangle + \frac{2}{3} \rho k \right) + \frac{\partial}{\partial x_j} \left((\nu + \nu_T) \frac{\partial \langle u_i \rangle}{\partial x_j} \right) + \underbrace{\frac{\partial}{\partial x_j} \left(\nu_T \frac{\partial \langle u_j \rangle}{\partial x_i} \right)}_{cross\ \ term} + F_i^e \tag{5.16}$$

In this equation, it is possible to include the isotropic term of the turbulent kinetic energy k in the average pressure $\langle p \rangle$, by defining $P_T = \langle p \rangle + 2/3\ \rho k$. As there is no difference in the numerical procedures between P_T and $\langle p \rangle$ and since physically $k << \langle p \rangle$, from here on, $\langle p \rangle$ will be used as notation for P_T. Furthermore, for flows which are homogeneous with respect to a direction, as in shear flows for instance, the cross terms of the transposed velocity gradient of equation (5.16) cancel, which gives

$$\frac{d \langle u_i \rangle}{dt} = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \frac{\partial}{\partial x_j} \left((\nu + \nu_T) \frac{\partial \langle u_i \rangle}{\partial x_j} \right) + F_i^e \tag{5.17}$$

In channel flows considered further, this cross term is also rigorously nil. As the eddy viscosity characterises the turbulent properties of the flow, it is thus not constant and has to be estimated.

The mixing length model

The simplest (and historically the first) model to compute the eddy viscosity ν_T introduces the "mixing length" assumption, such that the eddy viscosity can be written

$$\nu_T = L_m \times u_t \tag{5.18}$$

where u_t denotes the typical velocity of the large eddies and L_m the mixing length. It represents the characteristic distance of large eddy diffusive action ([69], [47]) and can thus be approximated by the integral scale L_i. The third important assumption of the mixing length model is the balance between production and dissipation of the turbulent kinetic energy: the production rate P (per mass unit) is defined by [54]

$$P = -\langle u_i' u_j' \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} \tag{5.19}$$

Considering equation (5.14), it follows that

$$P = 2\nu_T \langle s_{ij} \rangle \langle s_{ij} \rangle \tag{5.20}$$

The dissipation rate ϵ is thus approximated by

$$\epsilon \approx 2\nu_T \langle s_{ij} \rangle \langle s_{ij} \rangle \tag{5.21}$$

Combining equations (5.2), (5.18) and (5.21), u_t is estimated by

$$u_t = L_m \sqrt{2 \langle s_{ij} \rangle \langle s_{ij} \rangle} \tag{5.22}$$

The eddy viscosity ν_T is thus modelled by

$$\nu_T = L_m^2 \sqrt{2 \langle s_{ij} \rangle \langle s_{ij} \rangle} \tag{5.23}$$

In its generalised form (5.23), the mixing length model is applicable to all turbulent flows [69], provided the mixing length L_m is known. There are many industrial important flows that have been studied extensively, so that the appropriate specifications of L_m are well established. The prime example is boundary-layer flows in aeronautical applications (see the left picture of figure 5.9). For environmental applications [95], the mixing length model also gives accurate results for free surface flows for instance (see the right picture of figure 5.9). The major drawback of

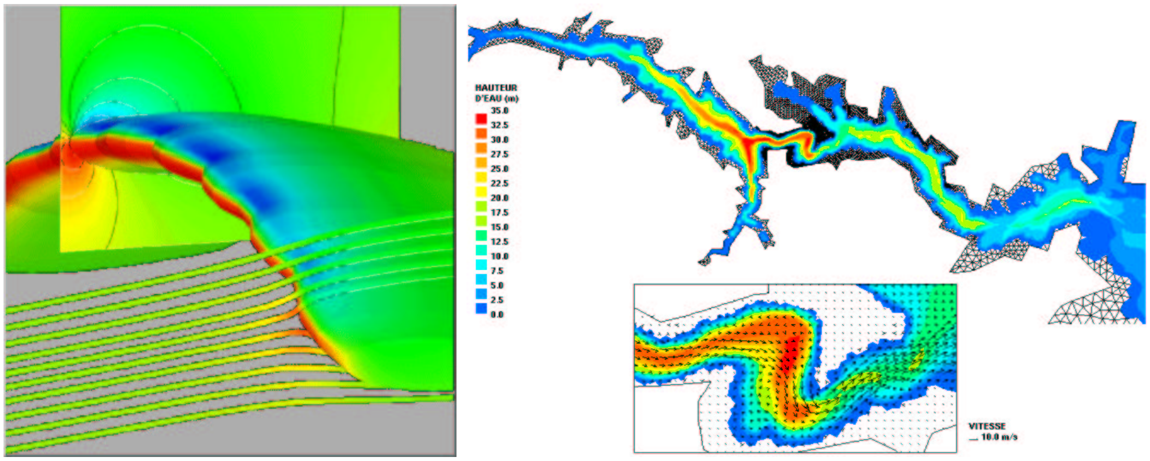

Figure 5.9: Pressure distribution of a 3D incompressible flow past a parafoil at Re=1 000 000 (left) [55] and Malpasset dam break simulated by Telemac-2D, with a mixing length model [31].

this model is its incompleteness: the mixing length L_m has to be specified, and the appropriate specification is inevitably dependent on the geometry of the flow. For a complex flow that has not been studied before, the specification of L_m requires a large measure of guesswork and consequently, one should have little confidence in the accuracy of the results. Morever, the assumption $P \approx \epsilon$ is not true for unsteady flows and close to walls [88]. Besides, in order to take into account the anisotropic effects of a turbulent flow, the scalar formulation of ν_T should be replaced by a tensor formulation [54].

The k model

The k model is also based on the eddy viscosity (5.14) and the mixing length assumptions (5.18). However, the velocity u_t is here estimated through the turbulent kinetic energy k by

$$u_t \propto \sqrt{k} \tag{5.24}$$

Since the mixing length L_m is close to the integral scale L_i, relation (5.2) gives

$$L_m \propto \frac{k^{3/2}}{\epsilon} \tag{5.25}$$

The eddy viscosity is thus expressed by

$$\nu_T = C_\mu \frac{k^2}{\epsilon} \tag{5.26}$$

where $C_\mu = 0.09$ [45]. In order to estimate the values of k and ϵ, a transport equation for k is solved. The k model hence belongs to the class of one-equation model. The transport equation for the turbulent kinetic energy is too complex to be directly solved [54] and has to be approximated. After modelling the diffusion and the production term of this equation [54], it is commonly written

$$\frac{\partial k}{\partial t} + \underbrace{\langle \underline{u} \rangle . \underline{\nabla} k}_{convection} = \underbrace{\underline{\nabla} . \left[\left(\nu + C_K \frac{k^2}{\epsilon} \right) \underline{\nabla} k \right]}_{diffusion} + \underbrace{P}_{production} - \underbrace{\epsilon}_{dissipation} \tag{5.27}$$

The constant C_K is equal to C_μ / σ_k where σ_k is a Schmidt number defined as the ratio between the eddy viscosity and the kinetic energy diffusion coefficient. σ_k is generally equal to 0.1. The production term P is usually modelled by

$$P = 2 C_\mu \frac{k^2}{\epsilon} \langle s_{ij} \rangle \langle s_{ij} \rangle \tag{5.28}$$

or

$$P = min\left(\sqrt{C_\mu}, 2C_\mu\frac{k}{\epsilon}\langle s_{ij}\rangle\langle s_{ij}\rangle\right)2k\langle s_{ij}\rangle\langle s_{ij}\rangle \tag{5.29}$$

Contrary to relation (5.28), equation (5.29) does not overestimate the turbulent kinetic energy k [25]. Indeed, it is linear with respect to the strain rate in case of rapid distortions. Once the k value is obtained, the value for ϵ is obtained through the following relation ([54])

$$\epsilon = C_\mu^{3/4}\frac{k^{3/2}}{L_m} \tag{5.30}$$

The values of k and ϵ are finally considered in (5.26). This model is well suited for unsteady flows, for which the energy production is characterised by a time scale $\tau_{L_i} \approx k/\epsilon$: contrary to the mixing length model, the k-model takes into account this transfer time. Once again, the drawback of this model is the mixing lenth definition requirement. This is avoided by using the following $k-\epsilon$ model.

The $k-\epsilon$ model

The $k-\epsilon$ model is also based on the eddy viscosity (5.14) and the mixing length assumptions (5.18). As the k model, the velocity u_t is estimated by the turbulent kinetic energy through (5.24) and the eddy viscosity ν_T by (5.26). The standard $k-\epsilon$ model, developed by Jones and Launder in 1972 [41], estimates the eddy vicosity through two transport equations, for k and ϵ. The $k-\epsilon$ model hence belongs to the class of two-equation models. The equation for k is identical to the equation (5.27) used for the k model and the production term is still modelled by (5.28). Similarly, the transport equation for ϵ is modelled by

$$\frac{\partial\epsilon}{\partial t} + \langle\underline{u}\rangle.\underline{\nabla}\epsilon = \underline{\nabla}.\left[C_\epsilon\frac{k^2}{\epsilon}\underline{\nabla}\epsilon\right] + \frac{\epsilon}{k}\left(C_{\epsilon 1}P - C_{\epsilon 2}\epsilon\right) \tag{5.31}$$

The five constants of the model are usually taken equal to [54]

$$\left\{\begin{array}{l} C_\mu = 0.09 \quad C_K = 0.09 \\ C_\epsilon = 0.069 \quad C_{\epsilon 1} = 1.44 \quad C_{\epsilon 2} = 1.92 \end{array}\right. \tag{5.32}$$

Near a wall, Dirichlet boundary conditions are usually used for k and ϵ according to

$$k = \frac{u_*^2}{\sqrt{C_\mu}} \quad and \quad \epsilon = \frac{u_*^3}{\kappa\delta} \tag{5.33}$$

where κ is the Karman constant and δ a small distance to the wall larger than the viscous sublayer thickness. The friction velocity u_* will be introduced in chapter 6. For a free surface, Neumann boundary conditions are generally considered

$$\frac{\partial k}{\partial \underline{n}} = 0 \quad and \quad \frac{\partial \epsilon}{\partial \underline{n}} = 0 \tag{5.34}$$

where $\underline{n}$ denotes the local normal vector to the free surface. By resolving the equation set relative to (5.27), (5.28) and (5.31), the values of k and ϵ are obtained.

Contrary to the two previous models, the mixing length definition is not required here. This model is thus self-sufficient since it is independent of the flow specifications. The standard $k-\epsilon$ is the most widely used turbulence model in commercial CFD codes and is a compromise between accuracy and simplicity (see figure 5.10). However, it does not give good results for stratified flows, for instance [54], or near zero-velocity point, even if some improvements have been made [25]. This model is also limited by the isotropic formulation of the eddy viscosity [44], [27]. Despite their broad range of applications, RANS models are sometimes not sufficient when rotation or 3D secondary motions are present, as it is the case for a turbulent flow in a square duct, for instance [83].

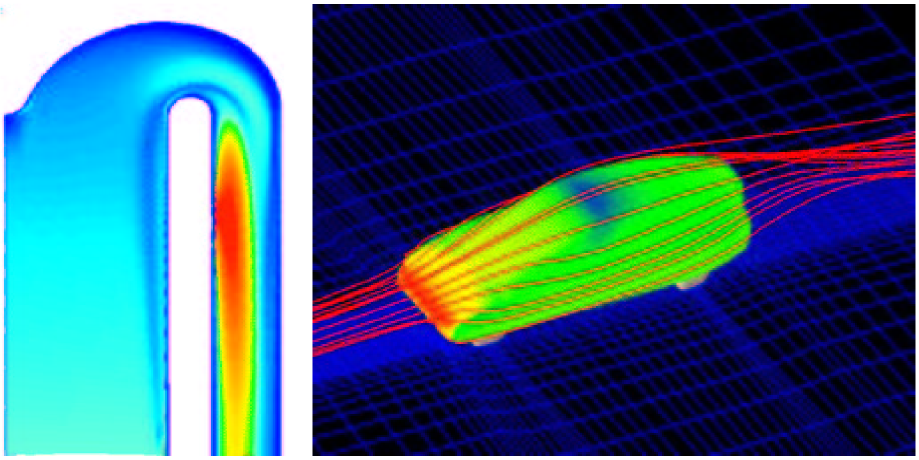

Figure 5.10: Turbulent viscosity countours in an axisymmetric bend [52] (left) and particle pathlines around a car and pressure shading on the car surface [53] (right), with a standard $k-\epsilon$ model.

5.2.3 Large Eddy Simulation (LES)

Principle

Large Eddy Simulation (LES) [48], [20] is motivated by the limitations of DNS and RANS models; the idea of LES is to simulate the large scales of motion of the tubulence while approximating the small ones. One can think of it as applying DNS to the large scales and RANS to the small scales. It is thus a compromise between the previous two approaches. The justification of such a treatment is that large eddies contain most of the energy, do most of the transporting of conserved properties and vary most from flow to flow. On the other hand, the small eddies are believed to be more universal [20] and less important, except for their dissipative effects easily modelled by an enhanced viscosity. Moreover, the large structures are mostly anisotropic whereas the small eddies are closer to isotropic [17].

Filtering operation

At first, it is essential to define modified fields that only contain the large scale components of total fields. Each flow variable Φ is thus decomposed into a large scale (or grid scale) component $\overline{\Phi}$ and a small scale (or subgrid scale) component Φ' such as

$$\Phi = \overline{\Phi} + \Phi' \tag{5.35}$$

The grid scale component is defined by the following filtering operator

$$\overline{\Phi}\left(\underline{x}, t\right) = \int_V G_\Delta\left(\underline{x} - \underline{x}'\right) \Phi\left(\underline{x}', t\right) d\underline{x}' \tag{5.36}$$

where $G_\Delta\left(\underline{x} - \underline{x}'\right)$ is a spatial filter function depending on the separation between the spatial vectors $\underline{x}$ and $\underline{x}'$. Δ corresponds to the filter size and V denotes the volume of the computation domain. The large scale field is hence a local average of the complete field. Filter functions which have been applied in LES include

- Gaussian function, represented on the left picture of figure 5.11 and defined by

$$G_\Delta(r) = \left(\frac{6}{\pi\Delta^2}\right)^{1/2} exp\left(-\frac{6r^2}{\Delta^2}\right) \tag{5.37}$$

  where r corresponds to the distance between $\underline{x}$ and $\underline{x}'$. The Gaussian filter has the advantage of being smooth.

- Box function, represented on the right picture of figure 5.11 and defined by

$$G_\Delta(r) = \frac{1}{\Delta} H\left(\frac{1}{2}\Delta - r\right) \tag{5.38}$$

  where H corresponds to the Heaviside function. This is simply an average over a rectangular region.

All these filters (for more details, see [69]) are considered as explicit filters. However, the nomenclature “subgrid scale” is derived from the kind of LES which considers the calculation grid as a filter. This type of filtering, refered as implicit filtering, is the simplest one and used in all basic LES models. Figure 5.12 shows a sample velocity $U(x)$ and the corresponding filtered velocity $\overline{U}(x)$ with the Gaussian filter [69]. At first, it is noticeable that $\overline{U}(x)$ follows the general trends of $U(x)$ and that the short lengthscale fluctuations have been removed.

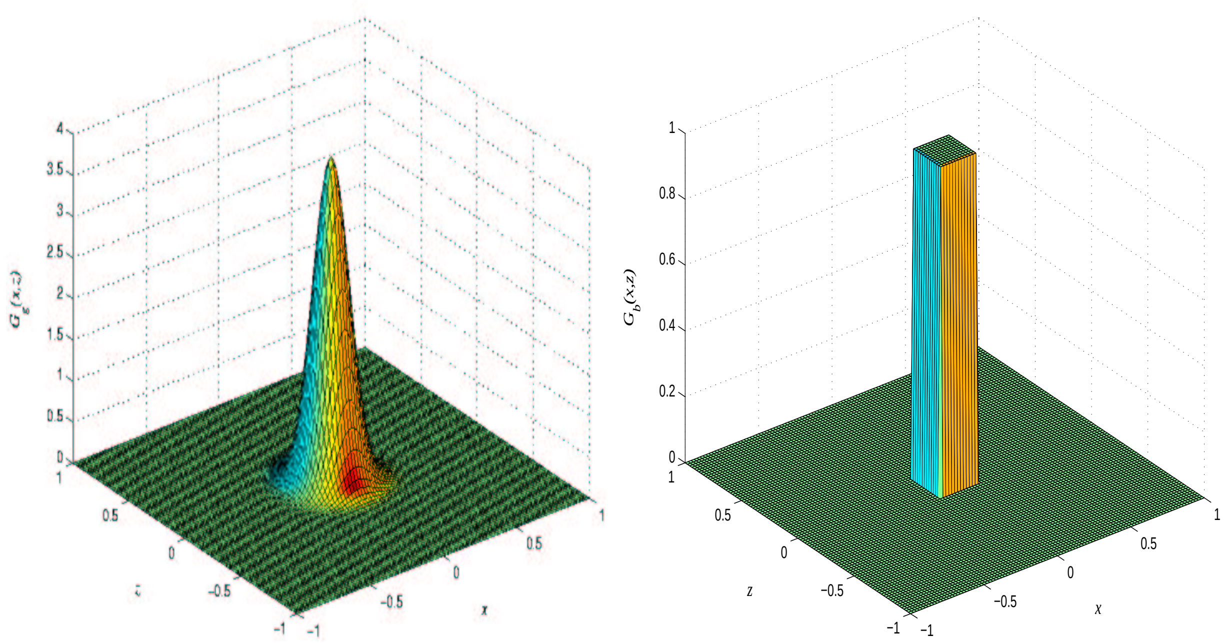

Figure 5.11: Gaussian (left) and box (right) filters.

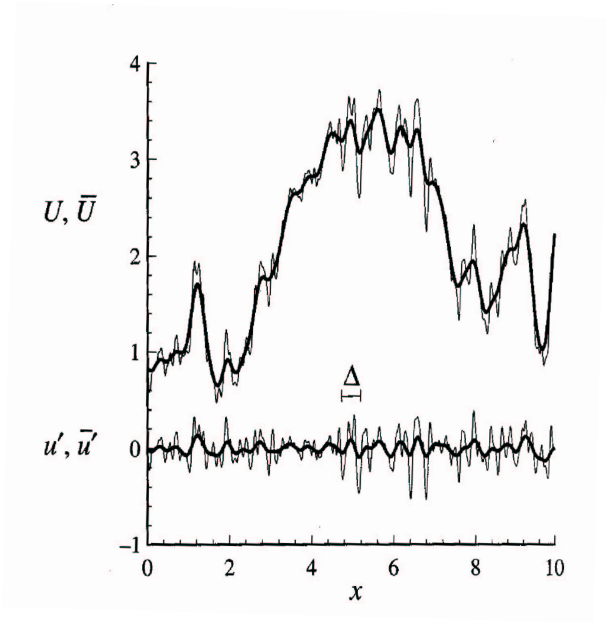

Figure 5.12: Upper curves: a sample of the velocity field $U(x)$ and the corresponding filtered field $\overline{U}(x)$ (bold line), using the Gaussian filter. Lower curves: the residual field $u'(x)$ and the filtered residual field $\overline{u'(x)}$ (bold line) [69].

These appear in the residual field $u'(x)$ which is also depicted in figure 5.12. Contrary to the Reynolds averaging previously introduced, figure 5.12 also reveals that the filtered subgrid scale component is non-zero. Another important difference between filtering and Reynolds averaging is that, in general, filtering a field a second time does not reproduce the original filtered field, i.e.

$$\overline{\overline{u}} \neq \overline{u} \tag{5.39}$$

Filtered incompressible Navier-Stokes equations

When filtering the incompressible Navier-Stokes equations, one obtains a set of equations very similar in form to the incompressible Reynolds equations (5.12) and (5.13):

$$\frac{\partial \overline{u_i}}{\partial x_j} = 0 \tag{5.40}$$

$$\frac{d\overline{u_i}}{dt} = \frac{\partial \overline{u_i}}{\partial t} + \overline{u}_j \frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + F_i^e \tag{5.41}$$

In the context of LES, τ_{ij} is called the subgrid scale tensor and is usually defined by

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j} \tag{5.42}$$

It plays a computational role in LES similar to the role played by the Reynolds stress in RANS models but the physics that it models is different, as one can notice from relation (5.39). Indeed, the subgrid scale energy is a much smaller part of the total flow than the RANS turbulent energy [20]. The subgrid scale tensor modelling may be hence less crucial than the Reynolds stress tensor modelling in RANS computations.

The Smagorinsky subgrid scale model

By far, the most commonly used subgrid scale model is the one proposed by Smagorinsky in 1963. It is based on an eddy viscosity assumption that can be considered as an adaptation, to the subgrid scale, of the Boussinesq assumption (5.14)

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -\nu_{Ts}\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) = -2\nu_{Ts}\overline{s_{ij}} \tag{5.43}$$

ν_{Ts} corresponds to a subgrid eddy viscosity and $\overline{s_{ij}}$ denotes the components of the rate of strain tensor based on filtered velocities

$$\overline{s_{ij}} = \frac{1}{2}\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) \tag{5.44}$$

The filtered momentum equation (5.41) is then written

$$\frac{d\overline{u_i}}{dt} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left((\nu + \nu_{Ts}) \frac{\partial \overline{u_i}}{\partial x_j} \right) + F_i^e \tag{5.45}$$

The Smagorinsky model is also based on a mixing length assumption, according to

$$\nu_{Ts} \propto L_m \times u^* \tag{5.46}$$

where L_m and u^* respectively denote a typical lengthscale and velocity of the subgrid scales. Approximating L_m by the filter size Δ, equation (5.46) becomes

$$\nu_{Ts} \propto \Delta u^* \tag{5.47}$$

The third important assumption of the Smagorinsky model is the local balance between the production P_{sgs} and the dissipation ϵ_{sgs} of the subgrid scales. P_{sgs} is defined by

$$P_{sgs} = -\tau_{ij} \frac{\partial \overline{u_i}}{\partial x_j} \tag{5.48}$$

Considering equation (5.43), it follows that

$$P_{sgs} = 2\nu_{Ts}\overline{s_{ij}}\;\overline{s_{ij}} \tag{5.49}$$

The subgrid scale dissipation ϵ_{sgs} is thus approximated by

$$\epsilon_{sgs} \approx 2\nu_{Ts}\overline{s_{ij}}\;\overline{s_{ij}} \tag{5.50}$$

In order to estimate u^*, the fourth assumption of the Smagorinsky model is to consider that the mixing length is close to the integral lengthscale of the subgrid scales, i.e.

$$\Delta \propto \frac{u^{*3}}{\epsilon_{sgs}} \tag{5.51}$$

The combination of equations (5.51), (5.47) and (5.50) provides the following estimation of u^*

$$u^* \propto \Delta\sqrt{2\overline{s_{ij}}\;\overline{s_{ij}}} \tag{5.52}$$

The Smagorinsky model expresses then the subgrid eddy viscosity ν_{Ts} according to

$$\nu_{Ts} = (C_S\Delta)^2 \sqrt{2\overline{s_{ij}}\;\overline{s_{ij}}} \tag{5.53}$$

The Smagorinsky constant C_S is generally equal to 0.1 and the parameter Δ is often equal to $2\,(\Delta x \times \Delta y \times \Delta z)^{1/3}$, where Δx (respectively Δy, Δz) denotes a characteristic lengthscale of the grid in x (resp. y, z) direction. From a formalistic point of view, the Smagorinsky model is identical to a statistical mixing length model with a constant mixing length. This is justified by the isotropic characteristic of the small eddies.

This basic Smagorinsky model generally gives good results, as shown in figures 5.13 and 5.14. In computational expense, it lies between RANS models and DNS [69]. The main two drawbacks of this model is general over dissipation. Moreover, it gives an incorrect estimation of the eddy viscosity near the walls [54], [69], [17].

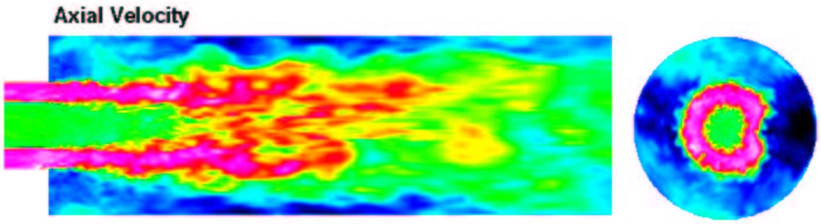

Figure 5.13: Axial velocity of a turbulent coaxial jet (left) by K. Akselvoll (Stanford University), with LES.

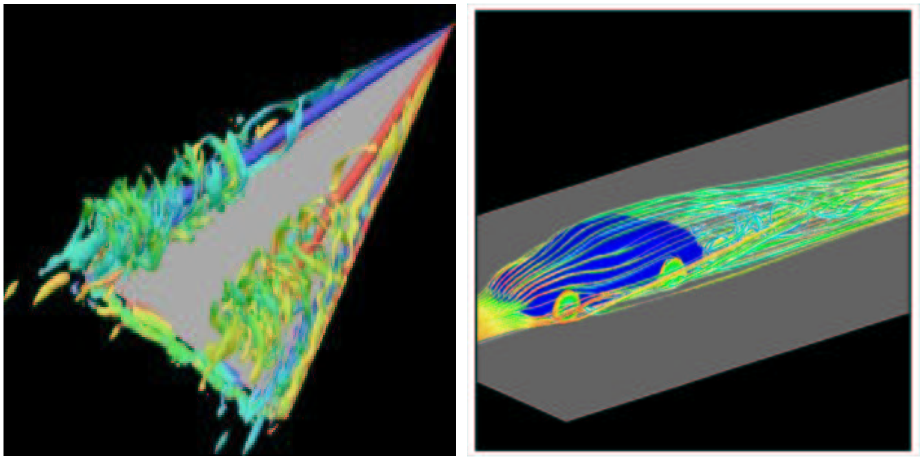

Figure 5.14: Turbulent separation on a Delta wing (left) (Onera) and velocity streamlines around an automobile by T*AFSM researchers (right), with LES.

5.3 Statistical turbulence modelling with Spartacus-2D

Currently, little research concerning turbulence modelling through SPH has been performed ([33], [96], [51], [94]): due to the Lagrangian characteristic of SPH, some stochastic models ([93], [96]) have first been developed. However, they are quite complex, specially regarding the boundary conditions. Three statistical RANS models, developed by Violeau since 2001 and implemented in Spartacus-2D, are presented herein.

5.3.1 Reynolds equations in Spartacus-2D

Since incompressible (or nearly incompressible) flows are considered in this work, equations (5.12) and (5.17) are used to derive a SPH form of the Reynolds equations with the eddy viscosity assumption. From a formal point of view, equations (5.12) and (5.17) are identical to the classical Navier-Stokes equations (2.44) (see part 2.4.1) and (2.50) (see part 2.4.2), except that

- pressure and velocities are considered as averaged values

- the kinematic viscosity is increased by the eddy viscosity

Taking into account these two differences, the Reynolds equations written in SPH formalism are thus identical to the SPH Navier-Stokes equations (2.45) and (2.79), implemented in Spartacus-2D. Consequently, for particle a, the averaged continuity equation considered in Spartacus-2D is written

$$\frac{d\rho_a}{dt} = \sum_b m_b \langle \underline{u} \rangle_{ab} \nabla_a w_h\left(r_{ab}\right) \tag{5.54}$$

with $\rho_a \approx \langle \rho \rangle_a$. $\langle \underline{u} \rangle_a$ corresponds to the averaged velocity relative to particle a and $\nabla_a w_h\left(r_{ab}\right)$ denotes the kernel gradient with respect to a-coordinates. In order to avoid confusion with the dissipation rate ϵ, the vector $\underline{\epsilon}_{ab}$ introduced in chapter 2 will not be used. Similarly and by considering the viscous term modelling (3.5), the averaged momentum equation implemented in Spartacus-2D is

$$\frac{d\langle \underline{u} \rangle_a}{dt} = -\sum_b m_b \left(\frac{\langle p \rangle_a}{\rho_a^2} + \frac{\langle p \rangle_b}{\rho_b^2} - 8\frac{\nu_{T,a} + \nu_{T,b}}{\rho_a + \rho_b} \frac{\langle \underline{u} \rangle_{ab} . \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \nabla_a w_h\left(r_{ab}\right) + \underline{F}_a^e \tag{5.55}$$

where $\nu_{T,a}$ corresponds to the eddy viscosity for particle a. The averaged pressure $\langle p \rangle_a$ is defined by the following filtered state equation

$$\langle p \rangle_a = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho_a}{\rho_0} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right] \tag{5.56}$$

The other notations used for these SPH equations are identical to those introduced in chapter 2. One can notice that the previous SPH equations are identical from a formalistic point of view to those described in chapter 3.

5.3.2 Mixing length model in Spartacus-2D

Considering the mixing length model (5.23), the eddy viscosity $\nu_{T,a}$ for each fluid particle a is expressed by

$$\nu_{T,a} = L_{m,a}^2 \sqrt{2 \langle s_{ij} \rangle_a \langle s_{ij} \rangle_a} \tag{5.57}$$

where

$$\langle s_{ij} \rangle_a = \frac{1}{2} \left[\left. \frac{\partial \langle u_i \rangle}{\partial x_j} \right|_a + \left. \frac{\partial \langle u_j \rangle}{\partial x_i} \right|_a \right] \tag{5.58}$$

$L_{m,a}$ corresponds to the value of the mixing length at the position occupied by particle a. In Spartacus-2D [94], the velocity gradient for particle a is estimated by

$$\left. \frac{\partial \langle u_i \rangle}{\partial x_j} \right|_a = -\frac{1}{\rho_a} \sum_b \underbrace{m_b \langle \underline{u} \rangle_{ab} \otimes \nabla_a w_h \left(r_{ab} \right)}_{A} \tag{5.59}$$

Since the A term of equation (5.59) is symmetric with respect to a and b subscripts, the symmetrical optimization technique introduced in part 2.2.3 is used to compute (5.59).
With this SPH mixing length model, the Spartacus-2D algorithm presented in figure 3.4 is modified by the computation of the eddy viscosiy for each fluid particle a according to

1. computation of the velocity gradient for each fluid particle a with (5.58)

2. computation of the eddy viscosity $\nu_{T,a}$ for each fluid particle a with (5.57)

This model has been applied by Violeau to a 2D channel [94] and a deeper investigation of a 2D free surface channel will be presented in chapter 6.

5.3.3 k model in Spartacus-2D

With a SPH k model, the eddy viscosity $\nu_{T,a}$ for each fluid particle a is expressed by

$$\nu_{T,a} = C_\mu \frac{k_a^2}{\epsilon_a} \tag{5.60}$$

where k_a and ϵ_a respectively correspond to the turbulent kinetic energy and the dissipation rate for particle a. In Spartacus-2D, Violeau [89] considers the following k-transport equation

$$\frac{dk_a}{dt} = \underbrace{\sum_b \frac{m_b}{\rho_b} \frac{\nu_{T,a} + \nu_{T,b}}{\sigma_k} \frac{k_{ab} \underline{r}_{ab}}{r_{ab}^2 + \eta^2} . \nabla_a w_h \left(r_{ab} \right)}_{diffusion} + \underbrace{P_a}_{production} - \underbrace{\epsilon_a}_{dissipation} \tag{5.61}$$

where $k_{ab} = k_a - k_b$. The diffusion term was written in the same form as the viscous diffusion term (2.78) developed by Monaghan, as Cleary did for the temperature conductivity [13].

The parameter σ_k corresponds to the Schmidt number and is equal here to 1.0 [89], [92]. The production term P_a can be modelled according to

$$P_a = 2C_{\mu_a}\frac{k_a^2}{\epsilon_a}\langle s_{ij}\rangle_a\langle s_{ij}\rangle_a \tag{5.62}$$

or

$$P_a = min\left(\sqrt{C_\mu}, 2C_\mu\frac{k_a}{\epsilon_a}\langle s_{ij}\rangle_a\langle s_{ij}\rangle_a\right)2k_a\langle s_{ij}\rangle_a\langle s_{ij}\rangle_a \tag{5.63}$$

Once again, the velocity gradient involved in $\langle s_{ij}\rangle_a$ definition are computed according to relation (5.59). To close the system, the condition (5.30) for ϵ_a is used, according to

$$\epsilon_a = C_\mu^{3/4}\frac{k_a^{3/2}}{L_{m,a}} \tag{5.64}$$

The other notations used for these SPH equations are identical to those introduced in chapter 2.

With this SPH k model, the Spartacus-2D algorithm presented in figure 3.4 is modified according to:

1. computation of the r.h.s. of (5.61) at the time n by considering (among others) (5.63) and (5.64)

2. integration of (5.61) and determination of k_a at the time $n+1$

3. determination of ϵ_a at the time $(n+1)$ with (5.64)

4. computation of the eddy viscosity $\nu_{T,a}$ at the time $(n+1)$ with (5.60)

This k model gives accurate and smooth results in a 2D free surface channel [89].

5.3.4 $k-\epsilon$ model in Spartacus-2D

With a SPH $k-\epsilon$ model, the eddy viscosity $\nu_{T,a}$ relative to each fluid particle a is also expressed by (5.60). The previous k transport equation (5.61) is unchanged and the production term modelling (5.63) is also maintained. In Spartacus-2D [89], the transport equation for ϵ is expressed by

$$\frac{d\epsilon_a}{dt} = \sum_b \frac{m_b}{\rho_b}\frac{\nu_{T,a}+\nu_{T,b}}{\sigma_\epsilon}\frac{\epsilon_{ab}\underline{r}_{ab}}{r_{ab}^2+\eta^2}\nabla_a w_h\left(r_{ab}\right) + \frac{\epsilon_a}{k_a}\left(C_{\epsilon 1}P_a - C_{\epsilon 2}\epsilon_a\right) \tag{5.65}$$

with $\epsilon_{ab} = \epsilon_a - \epsilon_b$. The constants used in (5.65) are equal to the value set (5.32) and $\sigma_\epsilon = 1.3$. Near a wall, k-Dirichlet boundary conditions are used according to

$$k_a = \frac{u_{*,a}^2}{\sqrt{C_\mu}} \tag{5.66}$$

where $u_{*,a}$ denotes the friction velocity for particle a (see chapter 6). For the ϵ-equation, Violeau uses Neumann boundary conditions, as described in [92].

With this SPH $k - \epsilon$ model, the Spartacus-2D algorithm presented in figure 3.4 is modified according to

1. computation of the r.h.s. of (5.61) and (5.65) at the time n by considering (among others) (5.63)
2. integral integration of (5.61) (respectively (5.65)) and determination of k_a (resp. ϵ_a) at the time $n + 1$
3. computation of the eddy viscosity $\nu_{T,a}$ at the time $(n + 1)$ with (5.60)

This SPH $k - \epsilon$ model has successfully been applied by Violeau [89] to a 2D free surface channel and a dam breaking.

5.4 Large Eddy Simulation with Spartacus-3D

One of the aims of this thesis is to adapt the concept of LES to SPH: in order to test a SPH Smagorinsky model, a 3D version of Spartacus-2D, Spartacus-3D, has been developed. Indeed, LES in 2D is not significant. Further information of this 3D-SPH code will be given in chapter 6.

5.4.1 LES filtering and SPH

In SPH formalism, the value of a function Φ for a particle a located at $\underline{x}$ is expressed by

$$\Phi\left(\underline{x}, t\right) = \int_\Omega \Phi\left(\underline{x'}, t\right) w_h\left(\underline{x} - \underline{x'}\right) d\underline{x'} \tag{5.67}$$

Moreover, as mentioned in part 5.2.3, filtered values $\overline{\Phi}$ in LES are obtained by the following convolution product

$$\overline{\Phi}\left(\underline{x}, t\right) = \int_V G_\Delta\left(\underline{x} - \underline{x'}\right) \Phi\left(\underline{x'}, t\right) d\underline{x'} \tag{5.68}$$

One can notice that these equations are very similar: the kernel function w_h is firstly analoguous to the filter function G_Δ. Secondly, the smoothing length h is equivalent to the filter size Δ. Consequently, it seems natural to consider LES in SPH.

5.4.2 Filtered Navier-Stokes equations in Spartacus-3D

Since incompressible (or nearly incompressible) flows are considered in this work, equations (5.40) and (5.45) are used to derive a SPH form of the filtered Navier-stokes equations with the eddy viscosity assumption. From a formalistic point of view, equations (5.40) and (5.45) are identical to the classical Navier-Stokes equations (2.44) (see part 2.4.1) and (2.50) (see part 2.4.2), except that

- pressure and velocities are considered as filtered values
- the kinematic viscosity is increased by the subgrid eddy visosity

Taking into account these two differences, the filtered Navier-Stokes equations written in SPH formalism are thus identical to the SPH Navier-Stokes equations (2.45) and (2.79), introduced in chapter 3. Consequently, for particle a, the filtered continuity equation considered in Spartacus-3D is written

$$\frac{d\rho_a}{dt} = \sum_b m_b \overline{\underline{u}}_{ab} \nabla_a w_h \left(r_{ab}\right) \tag{5.69}$$

with $\rho_a \approx \overline{\rho}_a$, since the fluid is nearly incompressible. $\overline{\underline{u}}_a$ corresponds to the filtered velocity for particle a. Similarly, the filtered momentum equation implemented in Spartacus-3D is

$$\frac{d\overline{\underline{u}}_a}{dt} = -\sum_b m_b \left(\frac{\overline{p}_a}{\rho_a^2} + \frac{\overline{p}_b}{\rho_b^2} - 8\frac{\nu_{Ts,a} + \nu_{Ts,b}}{\rho_a + \rho_b} \frac{\overline{\underline{u}}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \nabla_a w_h \left(r_{ab}\right) + \underline{F}_a^e \tag{5.70}$$

where the filtered pressure $\overline{p}_a$ is defined by the following filtered state equation

$$\overline{p}_a = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho_a}{\rho_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \tag{5.71}$$

Once again, the other notations used for these SPH equations are identical to those introduced in chapter 2.

5.4.3 Smagorinsky model in Spartacus-3D

Considering the Smagorinsky model (5.53), the eddy viscosity $\nu_{Ts,a}$ for each fluid particle a is expressed by

$$\nu_{Ts,a} = \left(C_S \delta\right)^2 \sqrt{2 \overline{s_{ij}}_a \overline{s_{ij}}_a} \tag{5.72}$$

where

$$\overline{s_{ij}}_a = \frac{1}{2} \left[\left. \frac{\partial \overline{u_i}}{\partial x_j} \right|_a + \left. \frac{\partial \overline{u_j}}{\partial x_i} \right|_a \right] \tag{5.73}$$

C_S corresponds to the Smagorinsky constant introduced in part 5.2.3 and δ a numerical parameter for the spatial discretisation of the problem, for instance twice the smoothing length h. This means that the structures of size inferior to $2h$ are considered as small structures. Further details concerning this model and some first results will be presented in chapter 6.

Chapter 6

Turbulent incompressible flow simulations with Spartacus-2D and Spartacus-3D

Turbulent flows play a central part in environmental and industrial applications, as previously emphasized. However, little research relative to turbulence modelling through SPH has been performed until now. In order to validate the SPH mixing length model introduced in chapter 5, 2D turbulent free surface channel flows are here simulated with Spartacus-2D. Moreover, a first attempt to adapt Large Eddy Simulation to SPH is also presented: two applications performed by the Smagorinsky model implemented in Spartacus-3D are described.

6.1 Theory of 2D turbulent free surface channel flows

This part presents some theoretical and experimental results relative to turbulent incompressible flows in a 2D free surface channel (see figure 6.1). A statistical point of view is here considered and our attention is confined to a stationnary fully developed flow, in which velocity statistics no longer vary with x.

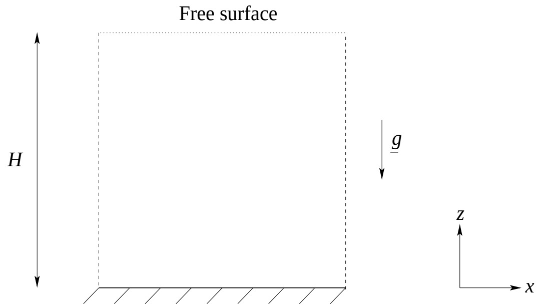

Figure 6.1: 2D free surface channel scheme.

6.1.1 General investigation

Definition of the total shear stress τ

For a 2D flow, the averaged continuity equation (5.12) is written

$$\frac{\partial \langle u \rangle}{\partial x} + \frac{\partial \langle v \rangle}{\partial z} = 0 \tag{6.1}$$

where $\langle u \rangle$ and $\langle v \rangle$ respectively denote the averaged axial and vertical velocity. Since $\langle u \rangle$ is independent of x and considering the boundary condition $\langle v \rangle = 0$ for $z = 0$, it follows that the averaged vertical velocity is zero within the flow. The averaged axial momentum equation (5.13) hence reduces to

$$0 = \nu \frac{d^2 \langle u \rangle}{dz^2} - \frac{d}{dz} \left\langle u^{'} v^{'} \right\rangle + F^e \tag{6.2}$$

where F^e corresponds to an horizontal forcing term per mass (pressure or channel bed slope) necessary to drive the fluid and $\langle u'v' \rangle$ denotes the axial Reynolds stress. This equation can also be written

$$\frac{d\tau}{dz} = -\rho F^e \tag{6.3}$$

where the total shear stress $\tau(z)$ is defined by

$$\tau(z) = \mu \frac{d \langle u \rangle}{dz} - \rho \langle u'v' \rangle \tag{6.4}$$

$\mu = \rho\nu$ corresponds to the dynamic viscosity of the fluid. The total shear stress $\tau(z)$ is thus the sum of the viscous stress $\mu \ d \langle u \rangle / dz$ and the Reynolds stress $\langle u'v' \rangle$. At the wall, the boundary condition $u = 0$ dictates that all Reynolds stress are zero. Consequently the wall shear stress τ_w is entirely due to the viscous contribution, i.e.

$$\tau_w = \mu \left. \frac{d \langle u \rangle}{dz} \right|_{z=0} \tag{6.5}$$

Viscous scales and the different layers

"Close" to the wall, the viscosity ν and the wall shear stress τ_w are important parameters in the case of smooth beds. From these quantities, one defines viscous scales that are the appropriate velocity scale and length scale in the near-wall region. These are the friction velocity u_*

$$u_* = \sqrt{\frac{\tau_w}{\rho}} \tag{6.6}$$

Layer	Location	Defining property
Viscous sublayer	$z^+ < 5$	Reynolds shear stress negligible compared with viscous stress
Buffer layer	$5 < z^+ < 30$	Region between viscous sublayer and log-law region
Log-law region	$z^+ > 30$	Log-law (6.22) holds
Outer layer	$z^+ > 50$	Direct effects of viscosity on $\langle u \rangle$ negligible

Table 6.1: The different layers and their defining properties [69].

and the viscous length δ_ν

$$\delta_\nu = \nu \sqrt{\frac{\rho}{\tau_w}} = \frac{\nu}{u_*} \tag{6.7}$$

The distance from the wall measured in viscous lengths, or wall units, is denoted by

$$z^+ = \frac{z}{\delta_\nu} = \frac{u_* z}{\nu} \tag{6.8}$$

Different layers, presented in table 6.1 [69], can then be defined on the basis of z^+.

6.1.2 Averaged axial velocity profiles

Considering the eddy viscosity assumption introduced in part 5.2.2, the averaged axial momentum equation (6.2) is written

$$\frac{d}{dz}\left[(\nu + \nu_T)\frac{d\langle u \rangle}{dz}\right] = constant \tag{6.9}$$

A first integration with respect to z gives

$$(\nu + \nu_T)\frac{d\langle u \rangle}{dz} = \frac{\tau_w}{\rho}\left(1 - \frac{z}{H}\right) \tag{6.10}$$

where τ_w denotes the wall shear stress (6.5) and H the channel height. The friction between the free surface and the atmosphere is here neglected. With the mixing length model presented in part 5.2.2, the eddy viscosity ν_T is here written

$$\nu_T = L_m^2 \frac{d\langle u \rangle}{dz} \tag{6.11}$$

By introducing the velocity scale u_* and the length scale δ_ν respectively defined by equations (6.6) and (6.7), one obtains

$$\nu_T^+ = \frac{\nu_T}{\nu} = L_m^+ \left(\frac{du^+}{dz^+}\right)^2 \tag{6.12}$$

where $u^+ = \langle u \rangle / u_*$ and $L_m^+ = L_m/\delta_\nu$. Equation (6.10) is then written

$$\left[1 + L_m^{+2}\left(\frac{du^+}{dz^+}\right)\right]\frac{du^+}{dz^+} = 1 - \frac{z^+}{H^+} \tag{6.13}$$

with $H^+ = H/\delta_\nu$.

Averaged axial velocity profile in the viscous sublayer

In the viscous sublayer ($z^+ < 5$), the large eddy size depends on the distance to the wall, as shown in figure 6.2 [88]. Indeed, close to the wall, it has experimentally been noticed that

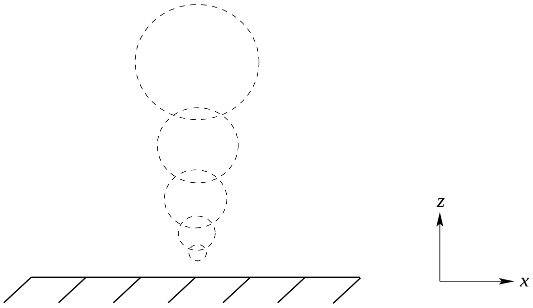

Figure 6.2: Evolution of the large eddy size in the viscous sublayer [88].

$$L_m \approx \kappa z \tag{6.14}$$

where $\kappa \approx 0.41$ is the Von Karman constant. Consequently, in the viscous sublayer

$$L_m^+ \frac{du^+}{dz^+} << 1 \tag{6.15}$$

and equation (6.13) hence becomes ([88])

$$\frac{du^+}{dz^+} \approx 1 - \frac{z^+}{H^+} \approx 1 \tag{6.16}$$

Therefore, the averaged axial velocity profile is linear in the viscous sublayer, i.e.

$$u^+ = z^+ \quad if \quad z^+ < 5 \tag{6.17}$$

Averaged axial velocity profile in the log-law layer

“Far” from the wall, the kinematic viscosity is much smaller than the eddy viscosity. Equation (6.13) hence gives

$$\left(L_m^+ \frac{du^+}{dz^+}\right)^2 = 1 - \frac{z^+}{H^+} \tag{6.18}$$

Moreover, a free surface damps the mixing length according to ([95])

$$L_m \approx \kappa z \sqrt{1 - \frac{z}{H}} \tag{6.19}$$

or, equivalently,

$$L_m^+ \approx \kappa z^+ \sqrt{1 - \frac{z^+}{H^+}} \tag{6.20}$$

Consequently,

$$\frac{du^+}{dz^+} = \frac{1}{\kappa z^+} \tag{6.21}$$

which gives by integration the log-law

$$u^+ = \frac{1}{\kappa} ln z^+ + B \tag{6.22}$$

or, equivalently,

$$\langle u \rangle (z) = u_* \left[\frac{1}{\kappa} ln \left(\frac{z u_*}{\nu} \right) + B \right] \tag{6.23}$$

where $B = 5.2 \pm 0.47$ [74].

6.1.3 Expression of characteristic terms

Eddy viscosity in the log-law layer

Considering equations (6.11), (6.19) and (6.21) gives the following evolution of the eddy viscosity in the log-law layer

$$\nu_T = \kappa u_* z \left(z - \frac{z}{h} \right) \tag{6.24}$$

Forcing term

Equation (6.9) could also be written

$$\frac{d}{dz} \left[(\nu + \nu_T) \frac{d \langle u \rangle}{dz} \right] = -F^e \tag{6.25}$$

Integrating (6.25) with respect to z and matching with (6.10) gives

$$F^e = \frac{u_*^2}{H} \tag{6.26}$$

Reynolds number

The Reynolds number is here based on the channel height H and the mean bulk velocity $\langle U \rangle$, defined by

$$\langle U \rangle = \frac{1}{H} \int_0^H \langle u \rangle (z) \, dz \tag{6.27}$$

Considering the averaged axial velocity (6.23) and relation (6.26), the Reynolds number is approximated, after an integration by parts, by

$$Re = \frac{\sqrt{F^e} H^{\frac{3}{2}}}{\nu} \left[\frac{1}{\kappa} ln \left(\frac{\sqrt{F^e} H^{\frac{3}{2}}}{\nu} \right) + B - \frac{1}{\kappa} \right] \tag{6.28}$$

where B and κ have been previously introduced.

6.1.4 Rough channel case

Up to this point, we have assumed that the channel bed was completely smooth. In fact, every surface is characterised by a length scale of protrusions ξ. For a fully rough bed, the roughness scale ξ is large compared with the viscous scale δ_ν, as represented in figure 6.3. Experimentally, this case corresponds to $\xi/\delta_\nu > 70$ [69]. The transfer of momentum from the fluid to the wall is then accomplished by the drag on the roughness elements. In this case, the averaged axial velocity profile can be approximated by

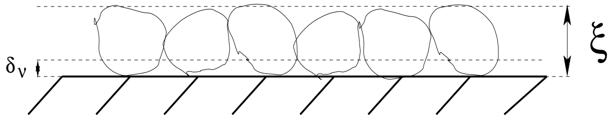

Figure 6.3: Fully rough wall scheme [88].

$$\langle u \rangle (z) = u_* \left[\frac{1}{\kappa} ln \frac{z}{\xi} + D \right] \tag{6.29}$$

with $D = 8.5$ [95].

6.2 Turbulent free surface channel flows simulated with Spartacus-2D[1]

The SPH mixing length model implemented in Spartacus-2D (see part 5.3.2) is here applied to the simulation of the outer layer ($z^+ > 50$) of turbulent free surface channel flows. Smooth and fully rough beds are successively considered.

[1] Submitted to "Journal of Hydraulic research"

Mean bulk velocity: $\langle U \rangle$ $(m.s^{-1})$	1.35
Reynolds number: $Re = \langle U \rangle H/\nu$	538 000
Friction velocity: u_* $(m.s^{-1})$	5.10^{-2}
Turbulent Reynolds number: $H^+ = u_* H/\nu$	20 000

Table 6.2: Physical characteristics of the considered turbulent free surface channel flow.

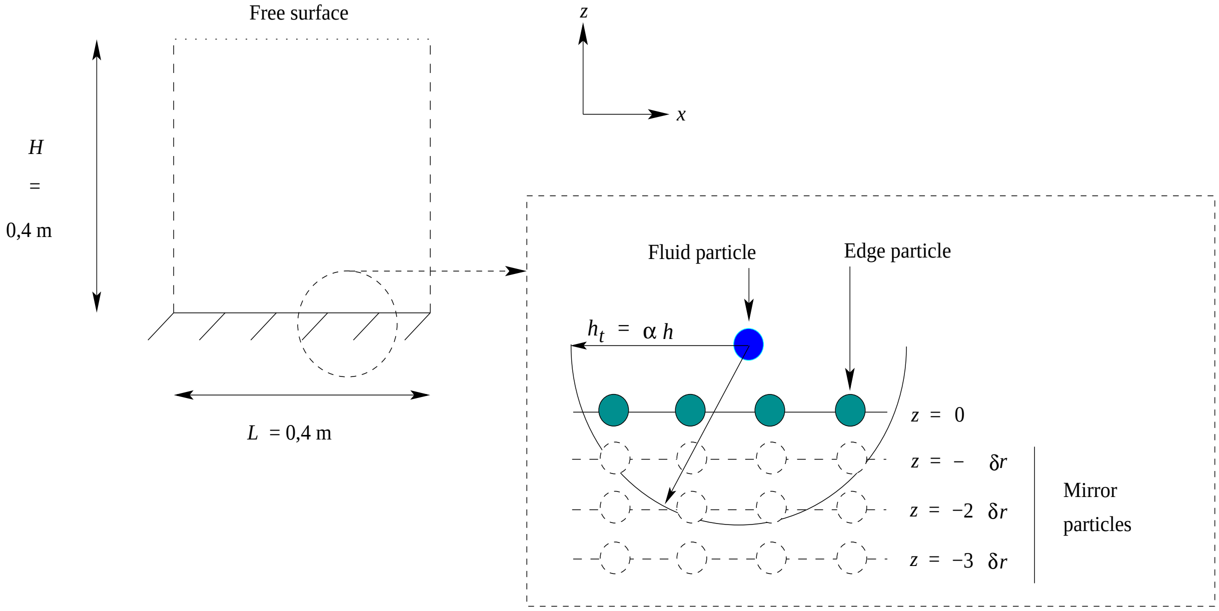

Figure 6.4: 2D turbulent free surface channel modelling (left) and zoom near the wall (right).

6.2.1 System modelling

Simulation conditions

The physical characteristics of the simulated flow are described in table 6.2. As explained in chapter 4, the fluid is driven by an horizontal external forcing term $\underline{F}_a^e$ applied to each fluid particle a at each time step. This external force is constant in space but not in time since it is updated at each time step to impose the correct mass flow rate (see part 3.1.3).

System discretisation and SPH wall function

The 2D free surface channel is firstly characterised by a smooth bed and modelled according to figure 6.4 with periodic conditions applied in the x-direction. The system discretisation is described in table 6.3:

- Edge particles represent fluid particles which are immediately located at the bottom of the outer layer. Their gravitational centre is hence placed at $z = \delta$, with $\delta^+ = 50$. The value of δ is thus defined by the considered friction velocity u_*. In some Eulerian

Fluid particles (x, z)	40×40
Edge particles	40
Mirror particles	120
Total particle number	1 760
Particle initial spacing: δr (cm)	1.

Table 6.3: Fluid, edge and mirror particle discretisation for the 2D turbulent free surface channel.

δ^+	50
Theoretical vertical position : $\delta(m)$	1.10^{-3}
Numerical vertical position (m)	0.

Table 6.4: Values relative to edge particle vertical position.

methods using cell vertex discretisation, although the first node is exactly located on the wall, the velocity at that point is non-zero and assumed to correspond to the value at position δ^+. This means that the flow domain is $[-\delta^+, H^+]$ and the computational domain $[0, H^+]$ (the wall is "pushed back"). The $-\delta^+$ offset is then negligible. This approximation is correct provided the turbulent Reynolds number H^+ (see table 6.2) is high enough. Table 6.4 describes the different values relative to edge particle position.

- As done in chapter 4, three layers of mirror particles are added under the edge particles, so that the fluid particles close to the wall do not suffer from a lack of neighbours.

Boundary conditions

The averaged density and pressure of mirror particles evolve at each time step as described in part 2.7.2, and they repel fluid particles from the wall through the averaged pressure gradient term. Moreover, the averaged axial velocity $\langle u_{edge} \rangle$ of edge particles must verify the log-law (6.23) for a smooth bed or equation (6.29) for a rough one. As a first approximation and since the friction velocity u_* is known in this case, $\langle u_{edge} \rangle$ can be numerically imposed at each time step according to

$$\langle u_{edge} \rangle = 0.747 \;\; m.s^{-1} \qquad for \;\; a \;\; smooth \;\; bed \tag{6.30}$$

or

$$\langle u_{edge} \rangle = 0.144 \;\; m.s^{-1} \qquad for \;\; a \;\; rough \;\; bed \tag{6.31}$$

However, in a more general way, the friction velocity and consequently the averaged boundary velocity can be estimated through the following procedure:

1. For each edge particle a, a fictitious point M located at a distance Δ from a on the normal to the wall (see figure 6.5) is defined.

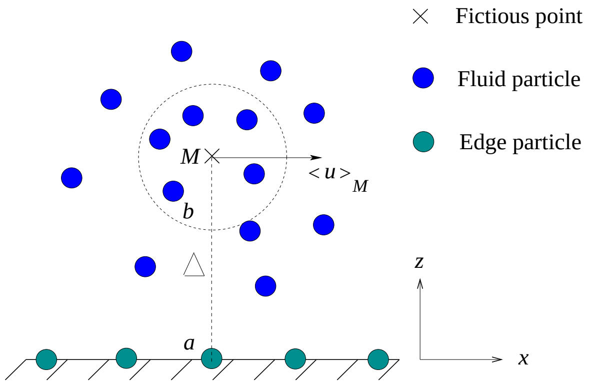

Figure 6.5: Computation of the averaged boundary velocity $\langle u_{edge} \rangle$.

2. The averaged axial velocity at M is computed with the classical following SPH relation

$$\langle u \rangle_M = \sum_b \frac{m_b}{\rho_b} \langle u \rangle_b w_h \left(r_{Mb} \right) \tag{6.32}$$

3. Since $\langle u \rangle_M$ should verify the following log-law for a smooth bed

$$\langle u \rangle_M = u_{*c} \left[\frac{1}{\kappa} ln \left(\frac{(\delta + \Delta) \, u_{*c}}{\nu} \right) + B \right] \tag{6.33}$$

   the friction velocity u_{*c} is then obtained by iteration. For a rough bed, the relation

$$\langle u \rangle_M = u_{*c} \left[\frac{1}{\kappa} ln \left(\frac{(\delta + \Delta)}{\xi} \right) + D \right] \tag{6.34}$$

   directly gives the value of u_{*c}.

4. The edge particle averaged axial velocity is computed according to

$$\langle u_{edge} \rangle = u_{*c} \left[\frac{1}{\kappa} ln \left(\frac{\delta u_{*c}}{\nu} \right) + B \right] \quad for \quad a \quad smooth \quad bed \tag{6.35}$$

   and

$$\langle u_{edge} \rangle = u_{*c} \left[\frac{1}{\kappa} ln \left(\frac{\delta}{\xi} \right) + D \right] \quad for \quad a \quad rough \quad bed \tag{6.36}$$

   It will be shown that the estimation of u_{*c} is very close to the theoretical value u_* in both cases.

Like the fluid particles, edge particles are transported at each time step with the averaged axial velocity $\langle u_{edge} \rangle$. This velocity is also directly applied to mirror particles, as shown in figure 6.6. This configuration, different from the one presented in part 4.2.3 (see figure 4.9), is the optimal one for velocity gradient estimation [38].

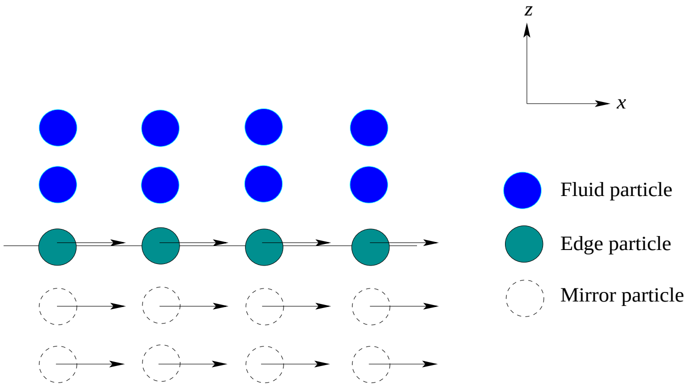

Figure 6.6: Edge particle averaged axial velocity directly applied to mirror particles.

Initial conditions

In order to start from a steady state, the initial particle distribution corresponds to the (40×40) steady state obtained in part 4.1. This is justified by the fact that the pressure at $z = \delta$ is nearly equal to the pressure at $z = 0$. Indeed, the discrepancy between the two values is less than 0.25 %. Also, as in chapter 4 (see part 4.2.2), fluid particles are initially characterised by a linear averaged axial velocity profile defined by

$$\langle u \rangle (z) = \langle u_{edge} \rangle - \left(\frac{z}{H} \right) \left[\langle u_{edge} \rangle - u_* \left(\frac{1}{\kappa} ln \left(\frac{H u_*}{\nu} \right) + B \right) \right] \quad (6.37)$$

and represented in figure 6.7.

SPH equations and numerical parameters

The velocity gradients relative to edge and fluid particles are determined with equation (5.59). The mixing length relative to this system is prescribed by equation (6.19) and the SPH mixing length model (5.57) is then used to compute the eddy viscosity of edge and fluid particles. Viscous effects are modelled by Monaghan's formulation and the gravity $\underline{g}$ and the driving force $\underline{F}^e$ are considered as external forces. The averaged momentum equation relative to particle a is therefore written

$$\frac{d\langle \underline{u} \rangle_a}{dt} = - \sum_b m_b \left(\frac{\langle p \rangle_a + \langle p \rangle_b}{\rho_a \rho_b} - 8 \frac{\nu_{T,a} + \nu_{T,b}}{\rho_a + \rho_b} \frac{\langle \underline{u} \rangle_{ab} . \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \nabla_a w_h \left(r_{ab} \right) + \underline{g} + \underline{F}_a^e \quad (6.38)$$

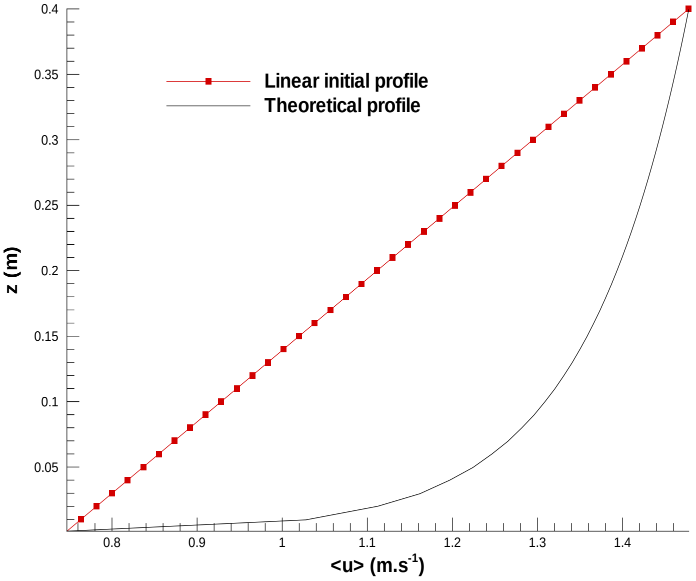

Figure 6.7: Initial averaged axial velocity profile.

Order of the spline kernel	4
$h/\delta r$	1.5
$c_0\ \left(m.s^{-1}\right)$	50

Table 6.5: Main SPH numerical parameters for the tubulent free surface channel flow.

and the averaged continuity equation (5.54) is used to compute the fluid and edge particle density. The averaged Navier-Stokes and the position equations are integrated in time with a fully explicit method (see part 3.3.2) and the averaged particle pressure is then determined through the averaged state equation (5.56). The main SPH numerical parameters relative to this test case are presented in table 6.5.

6.2.2 Simulation results relative to a smooth bed

Results for a simplified modelling

At first, the theoretical value of the averaged axial velocity gradient $d\langle u\rangle/dz$ (which is the main component of the averaged stress tensor $\langle s_{ij}\rangle$) is imposed, according to

$$\frac{d\langle u\rangle}{dz} = \frac{u_*}{\kappa z} \tag{6.39}$$

Moreover, the theoretical value of the averaged axial velocity of edge particles is also imposed. Figure 6.8 reveals that the averaged axial velocity field after convergence is very stable. Indeed, the particles remain organised and one can notice that the system is perfectly homogeneous in the x-direction. All profiles presented in the following are therefore obtained with a spatial averaging along x-direction. Figure 6.9 shows four averaged axial velocity profiles at different times: it reveals that the code converges around 30 s towards a stable velocity profile close to the theoretical one. During this calculation, a particle is transported a distance equal to 121 times the channel length, which is larger than the usual criteria relative to result convergence. The eddy viscosity is perfectly computed, as one can notice from figure 6.10. In order to consider a more general case, the boundary velocity relative to edge particles is now computed with the method previously described. Figure 6.11 proves that the computation of the boundary velocity is satisfactory.

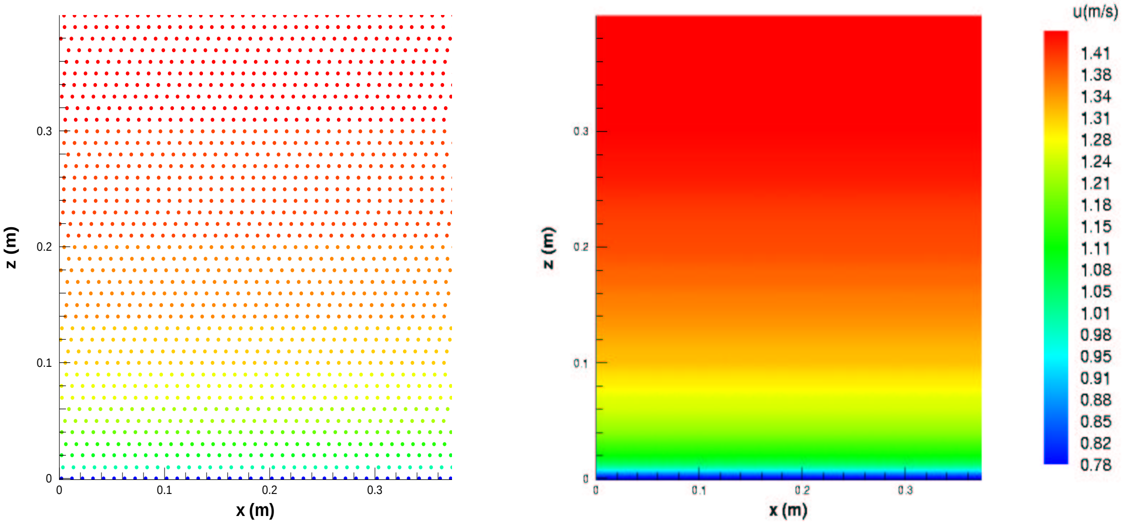

Figure 6.8: Particle position (left) and averaged axial velocity field (right) after convergence, with $d\langle u\rangle/dz$ and $\langle u_{edge}\rangle$ imposed to the theoretical values.

Results for a general modelling

All averaged velocity gradients are now computed with the relation (5.59) and the boundary velocity is also computed at each time step. The averaged velocity gradient $d\langle u\rangle/dz$ after convergence is represented in figure 6.12. One can notice that $d\langle u\rangle/dz$ is underestimated near the wall and its evolution is quite perturbed in the upper part of the channel. Comparison of figures 6.10 and 6.13 reveals that the noise present in the eddy viscosity profile of figure 6.13 is due to the small error associated to the evaluation of $d\langle u\rangle/dz$ by SPH: as the eddy viscosity involves $(d\left\langle u\right\rangle/dz)^2$, it is thus not surprising that the noise present in averaged

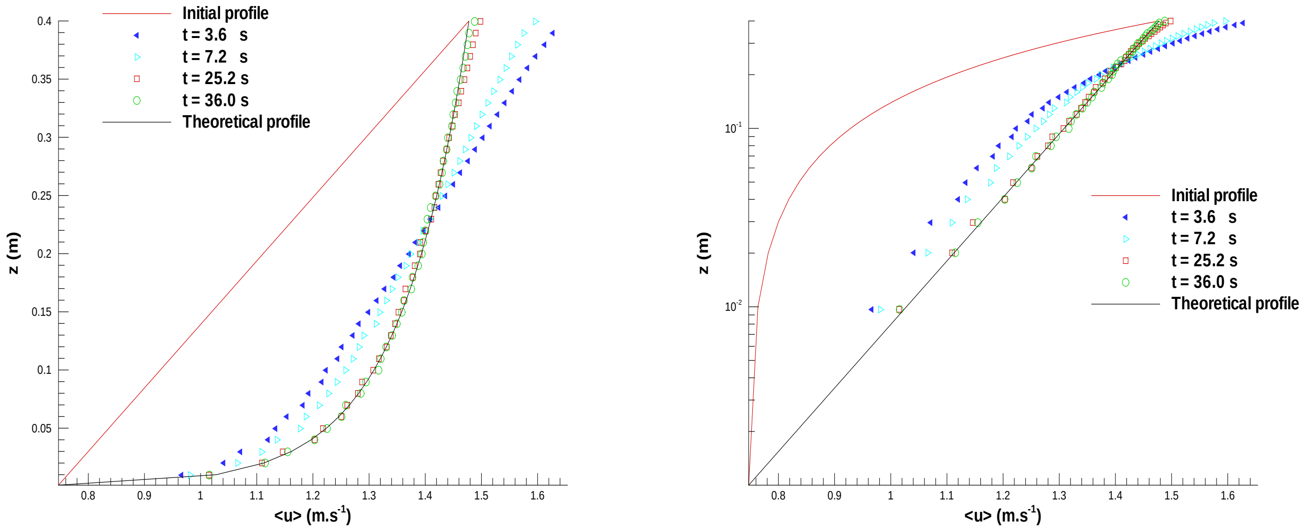

Figure 6.9: Averaged axial velocity profiles at different times, with $d\langle u\rangle/dz$ and $\langle u_{edge}\rangle$ imposed to the theoretical values. left: normal scale; right: logarithmic scale.

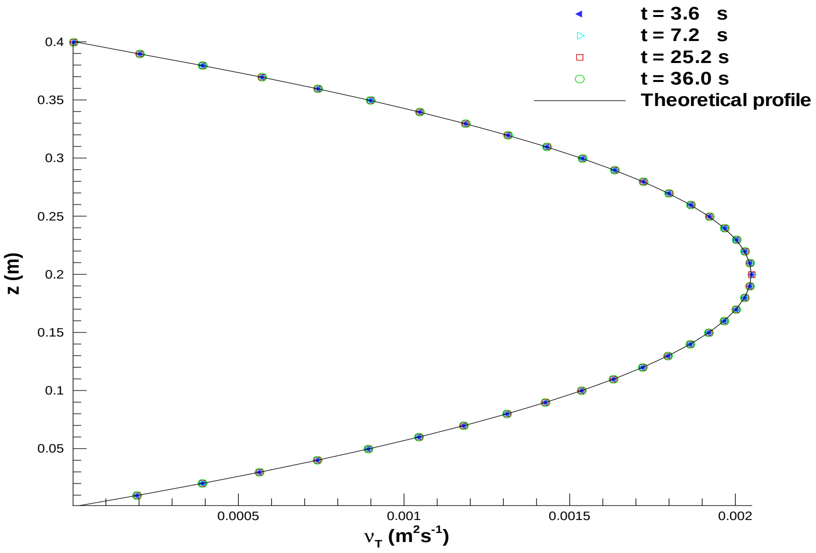

Figure 6.10: Eddy viscosity profiles after convergence, with theoretical values of $d\langle u\rangle/dz$ and $\langle u_{edge}\rangle$.

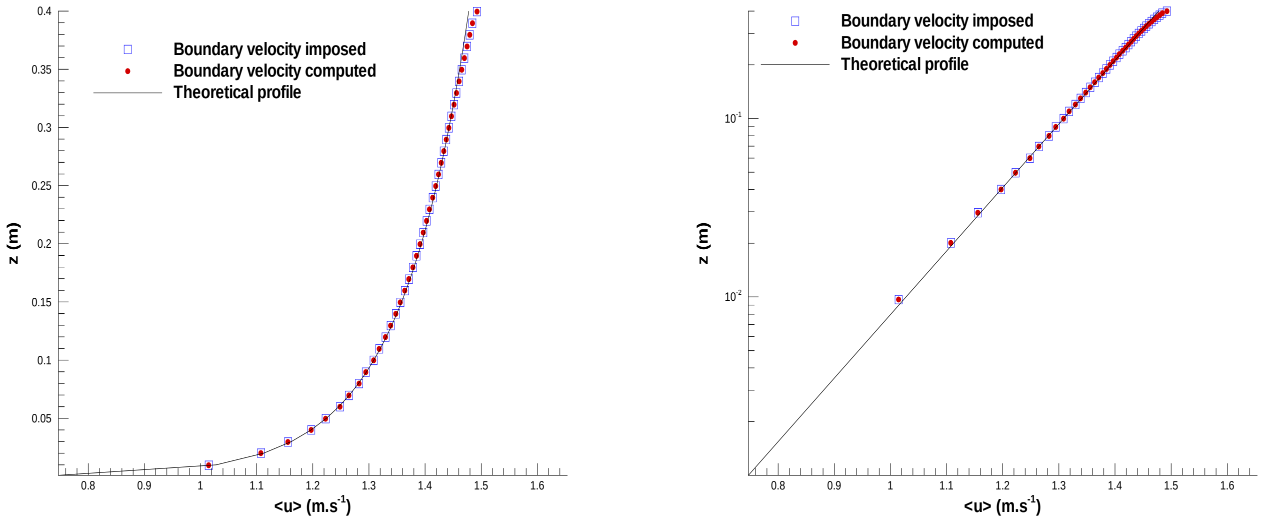

Figure 6.11: Averaged axial velocity profiles after convergence, with theoretical values of $d\langle u\rangle/dz$. left: normal scale; right: logarithmic scale.

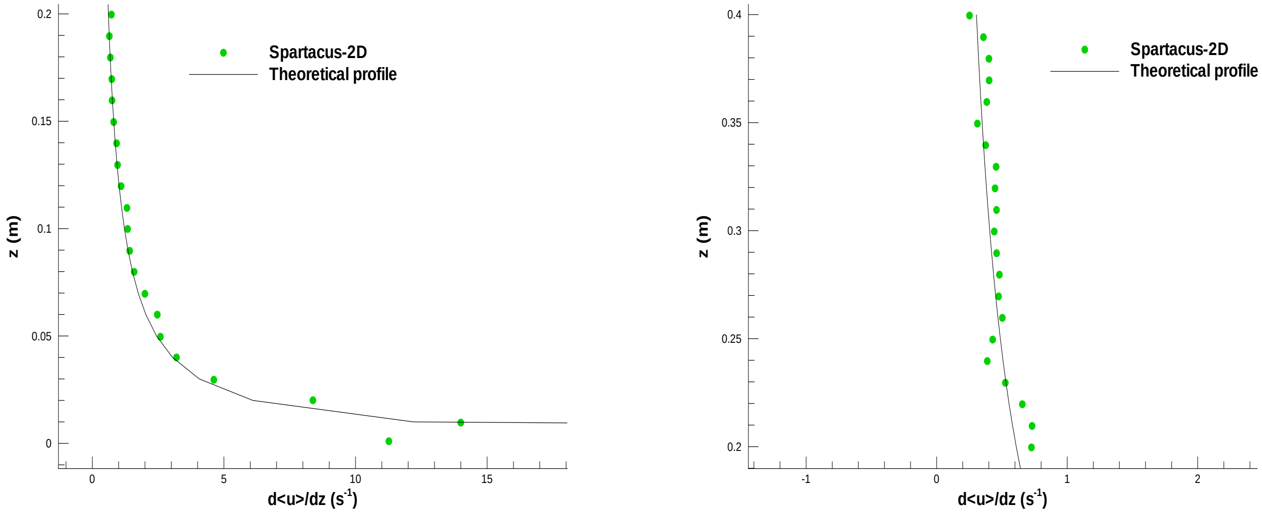

Figure 6.12: Averaged axial velocity gradient profiles in the lower (left) and the upper (right) part of the channel, after convergence.

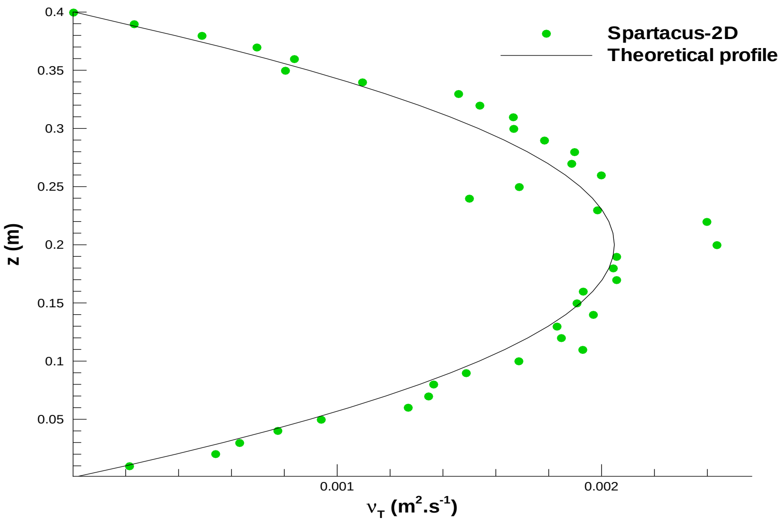

Figure 6.13: Eddy viscosity profiles after convergence.

axial velocity gradient is amplified in the eddy viscosity profile. It is actually difficult to perfectly reproduce strong averaged axial velocity gradient relative to a logarithmic velocity field ([38], [5]). However, the averaged axial velocity profile after convergence is quite close to the theoretical one, as shown in figure 6.14. All the profiles can be smoothed by a temporal averaging, achieved after convergence on a few time steps (see figure 6.15 and 6.16). The overestimation of the eddy viscosity is consistent with the slight overestimation of the averaged velocity gradient $d\langle u\rangle/dz$.

6.2.3 Simulation results relative to a fully rough bed

We now consider a rough bed channel characterised by a scale of protrusion ζ. Arbitrarily, we chose $\zeta = 1.10^{-2}$ m, which corresponds to a fully rough bed for the present Reynolds number. Indeed, the Reynolds number $Re_* = u_*\xi/\nu$ is here equal to 500, which is much higher than the critical value 70 [95]. Numerically, one of the differences with the previous smooth bed case is the boundary condition: velocity of edge particles is now computed with equation (6.36). Moreover, the mean bulk velocity is decreased if the friction velocity is maintained. In this case, $\langle U\rangle = 0.746\ m.s^{-1}$. The averaged velocity gradients and the boundary velocity are computed at each time step, as previously described. Considering a linear initial velocity profile similar to the one represented in figure 6.7, the profiles after convergence are close to the theoretical ones, as shown in figures 6.17 and 6.18.

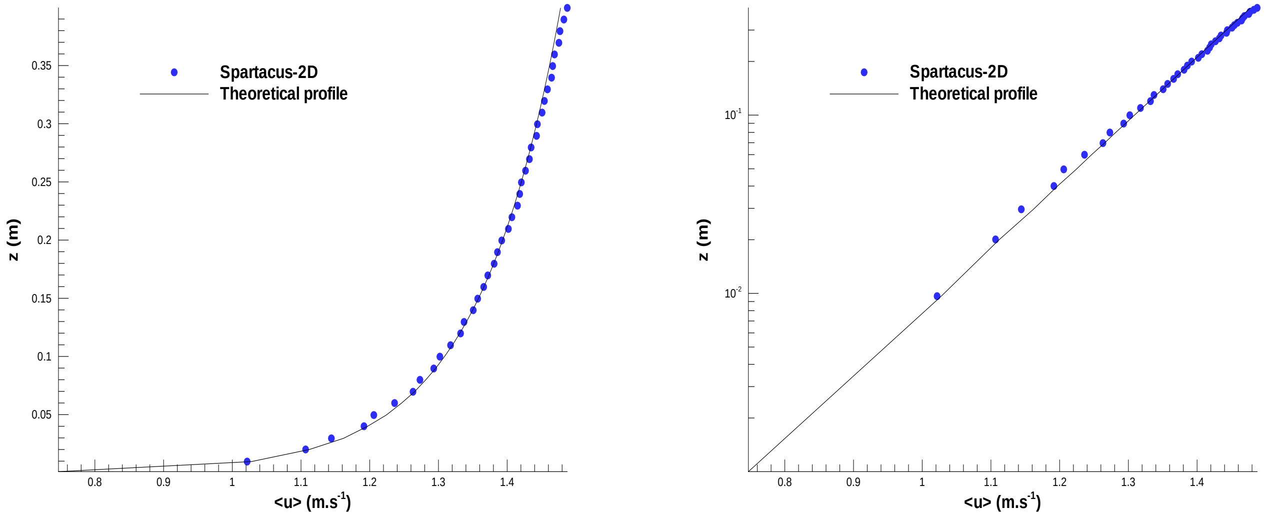

Figure 6.14: Averaged axial velocity profiles after convergence. left: normal scale; right: logarithmic scale .

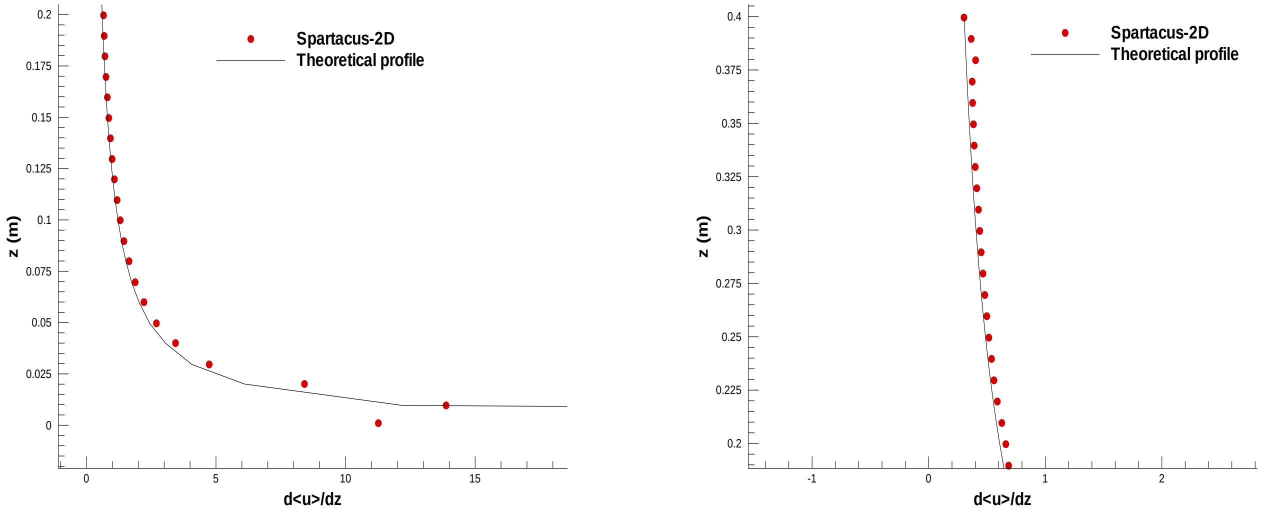

Figure 6.15: Averaged axial velocity gradient profiles in the lower (left) and the upper (right) part of the channel, with a temporal averaging achieved after convergence.

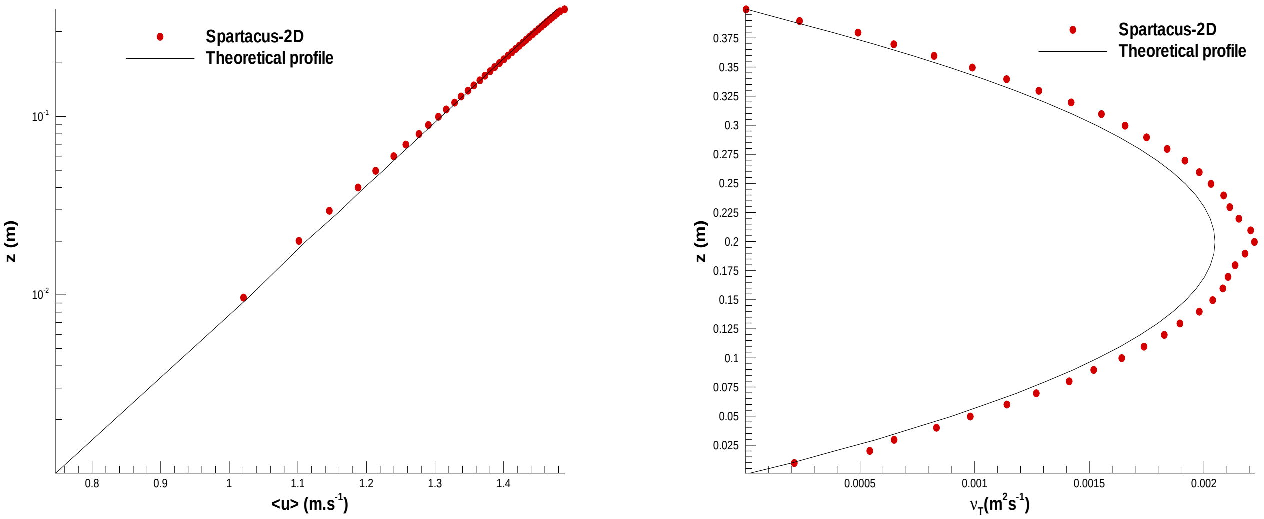

Figure 6.16: Averaged axial velocity and eddy viscosity profiles, with a temporal averaging achieved after convergence.

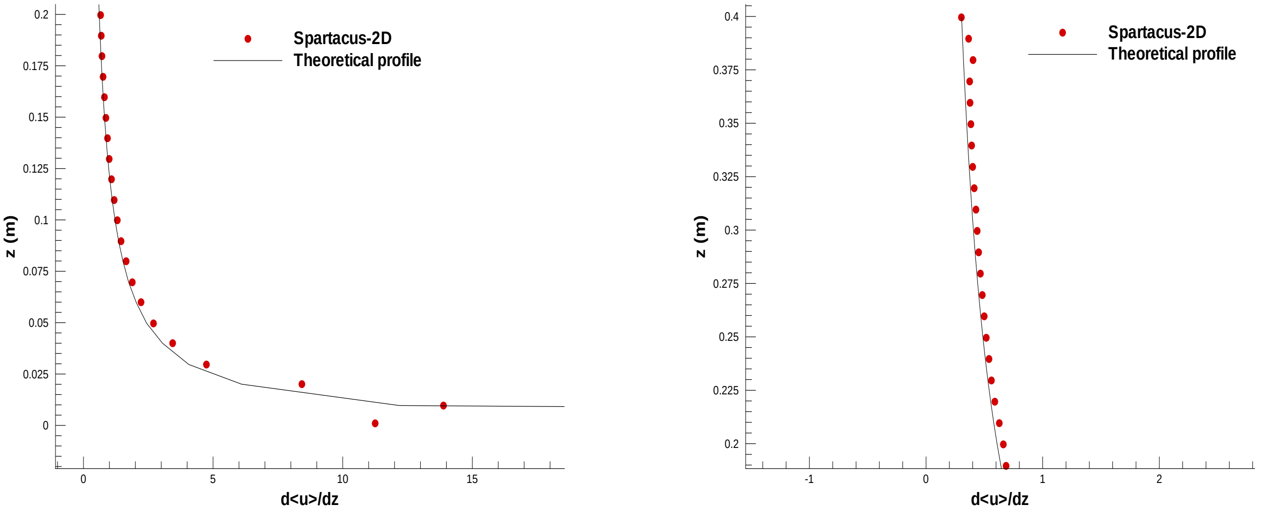

Figure 6.17: Averaged axial velocity gradient profiles in the lower (left) and the upper (right) part of a fully rough channel, with a temporal averaging achieved after convergence.

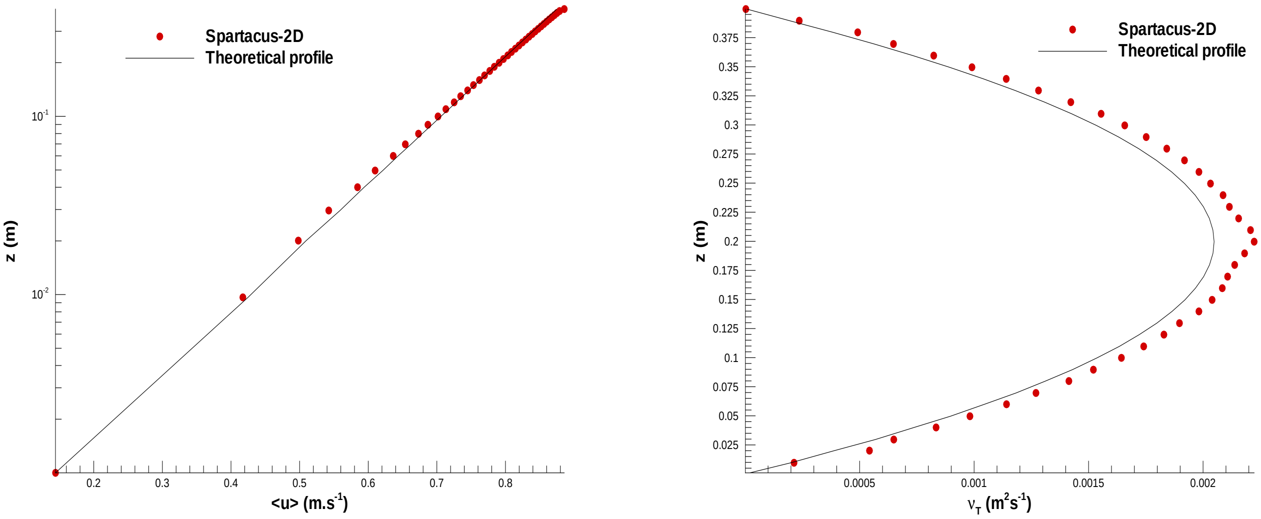

Figure 6.18: Averaged axial velocity and eddy viscosity profiles in a fully rough channel, with a temporal averaging achieved after convergence.

6.2.4 Partial conclusion

The SPH mixing length model gives satisfactory results for a 2D turbulent free surface channel characterised by smooth and rough beds: the particles remain organised and the averaged axial velocity field is close to the theoretical one. However, the velocity gradient estimation is quite irregular and slightly overestimated, which leads to overestimated eddy viscosity profiles.

6.3 The numerical code Spartacus-3D

The SPH code Spartacus-3D, based on Spartacus-2D, has been developed during this work to test the concept of Large Eddy Simulation (LES) in SPH. This section describes the fundamental LES equations implemented in Spartacus-3D and also presents its general algorithm structure.

6.3.1 Fundamental filtered equations

The filtered continuity equation

The filtered continuity equation implemented in Spartacus-3D corresponds to equation (5.69).

The filtered equation of motion

As in Spartacus-2D, two discretisations of the filtered pressure gradient term are implemented in Spartacus-3D:

$$\nabla \overline{p}_a = \sum_b \frac{m_b}{\rho_b} \left(\overline{p}_b + \overline{p}_a\right) \dot{w}_h \left(r_{ab}\right) \underline{e}_{ab} \tag{6.40}$$

and

$$\nabla \overline{p}_a = \rho_a \sum_b m_b \left(\frac{\overline{p}_b}{\rho_b^2} + \frac{\overline{p}_a}{\rho_a^2}\right) \dot{w}_h \left(r_{ab}\right) \underline{e}_{ab} \tag{6.41}$$

We recall that $\rho_a \approx \overline{\rho}_a$, since nearly incompressible flows are considered in this work.
Only the viscous term (3.5) developed by Monaghan is considered in Spartacus-3D. Indeed, the formulation (2.77) established by Morris is computationally inefficient for turbulent flows. The filtered viscous term is then expressed by

$$\sum_b \frac{8}{\rho_a + \rho_b} \left(\nu_{T,a} + \nu_{T,b} + 2\nu\right) \frac{\overline{\underline{u}}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \dot{w}_h \left(r_{ab}\right) \underline{e}_{ab} \tag{6.42}$$

where $\nu_{T,a}$ denotes the eddy viscosity for particle a and ν the kinematic viscosity of the fluid. An example of the filtered momentum equation implemented in Spartacus-3D is given by equation (5.70), where the kinematic viscosity ν has been neglected.

SPH Smagorinsky model

The eddy viscosity is computed through the SPH Smagorinsky model (5.72) and the filtered velocity gradients estimated with equation (5.73). The value of the Smagorinsky constant C_S is adjusted by the user in a parameter file.

External forces

All external forces that could be considered in Spartacus-2D (see part 3.1.3) could also be included in Spartacus-3D in the same way.

The filtered state equation

Filtered particle pressure is determined through the filtered state equation (5.71).

Time step and temporal integration

As in Spartacus-2D, the time step δt is here evaluated through the condition (3.23) and the previous filtered equations are integrated in time with the fully explicit method.

6.3.2 Kernel and averaged values

Kernel derivative

As in Spartacus-2D, the first derivative of the third, fourth and fifth order spline kernels are considered in Spartacus-3D, where they are defined by

$$\dot{w}_h\left(q\right) = \frac{1}{\pi h^3} \begin{cases} -3q + \frac{9}{4}q^2 & \text{if} \quad 0 \leq q \leq 1 \\ -\frac{3}{4}\left(2-q\right)^2 & \text{if} \quad 1 \leq q \leq 2 \\ 0 & \text{if} \quad q \geq 2 \end{cases} \tag{6.43}$$

$$\dot{w}_h\left(q\right) = \frac{51}{2000h^3} \begin{cases} -4\left(\frac{5}{2}-q\right)^3 + 20\left(\frac{3}{2}-q\right)^3 - 40\left(\frac{1}{2}-q\right)^3 & \text{if} \quad 0 \leq q \leq 0.5 \\ -4\left(\frac{5}{2}-q\right)^3 + 20\left(\frac{3}{2}-q\right)^3 & \text{if} \quad 0.5 \leq q \leq 1.5 \\ -4\left(\frac{5}{2}-q\right)^3 & \text{if} \quad 1.5 \leq q \leq 2.5 \\ 0 & \text{if} \quad q \geq 2.5 \end{cases} \tag{6.44}$$

$$\dot{w}_h\left(q\right) = \frac{7}{478\pi h^3} \begin{cases} -5\left(3-q\right)^4 + 30\left(2-q\right)^4 - 75\left(1-q\right)^4 & \text{if} \quad 0 \leq q \leq 1 \\ -5\left(3-q\right)^4 + 30\left(2-q\right)^4 & \text{if} \quad 1 \leq q \leq 2 \\ -5\left(3-q\right)^4 & \text{if} \quad q \geq 3 \\ 0 & \text{if} \quad q \geq 3 \end{cases} \tag{6.45}$$

with q denoting the ratio r_{ab}/h.

Averaging with the kernel

In order to obtain the averaged values of a LES, the classical process used in Eulerian methods is here adapted to SPH. The definition of a fixed grid at the beginning of the calculation is hence necessary. If a periodic flow is investigated, the initial particle distribution could be considered as a fixed grid. The idea to use a grid in a particle code was first introduced in PIC method (see chapter 1). In Spartacus-3D, the averaged values are obtained according to the following process.

1. Filtered fields are interpolated on the fixed grid with the kernel function, as represented in figure 6.19. In order to spare computing time, the cubic cell method described in part 6.3.3 is considered and the third order spline kernel, defined by

$$w_h\left(q\right) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if} \quad 0 \leq q \leq 1 \\ \frac{1}{4}\left(2-q\right)^3 & \text{if} \quad 1 \leq q \leq 2 \\ 0 & \text{if} \quad q \geq 2 \end{cases} \tag{6.46}$$

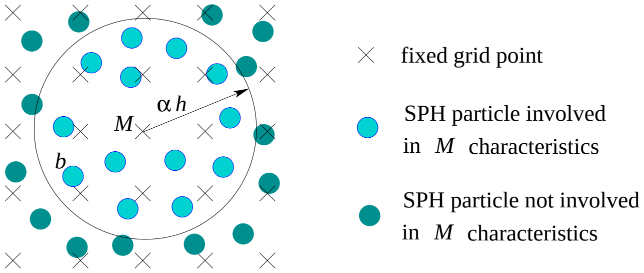

Figure 6.19: Interpolating filtered fields to the fixed grid with the kernel function.

is used. The filtered value $\overline{A}$ for the grid point M is thus estimated by

$$\overline{A}_M = \sum_b \frac{m_b}{\rho_b} \overline{A}_b w_h \left(r_{Mb} \right) \tag{6.47}$$

where b corresponds to SPH particles close to M. Two examples of interpolated fields are presented in figures 6.20 and 6.21.

2. For a periodic flow, all interpolated fields are temporally averaged until the end of the calculation. In classical Eulerian codes, the temporal averaging could be achieved at each time step. In Spartacus-3D, since the previous interpolating process is time consuming, the temporal averaging is carried out each 100 or 1000 time steps.

3. If the flow is characterised by one or several homogeneous directions, the interpolated fields are also spatially averaged along the homogeneous direction(s).

6.3.3 Optimisations and algorithm structure

Number of neighbours in a 3D approach

In a 2D approach, the neighbours for a particle a are located within a disc characterised by a radius proportional to the smoothing length h and centred on a. In a 3D approach, they are located within a sphere characterised by the same radius, which considerably increases the number of neighbours (see figure 6.22). For instance, considering a fourth order spline kernel and a smoothing length such as $h = 1.5\ \delta r$ gives 44 neighbours for a particle a in the case of the 2D free surface channel investigated in this work. In a 3D approach, the same parameters give 250 neighbours whereas a simple finite difference discretisation on a square grid only relates 12 nodes. Nowadays, workstations and computers can not handle such high number of neighbours in a SPH code. In order to reduce the CPU time, the user should consider a low order kernel and a small smoothing length value.

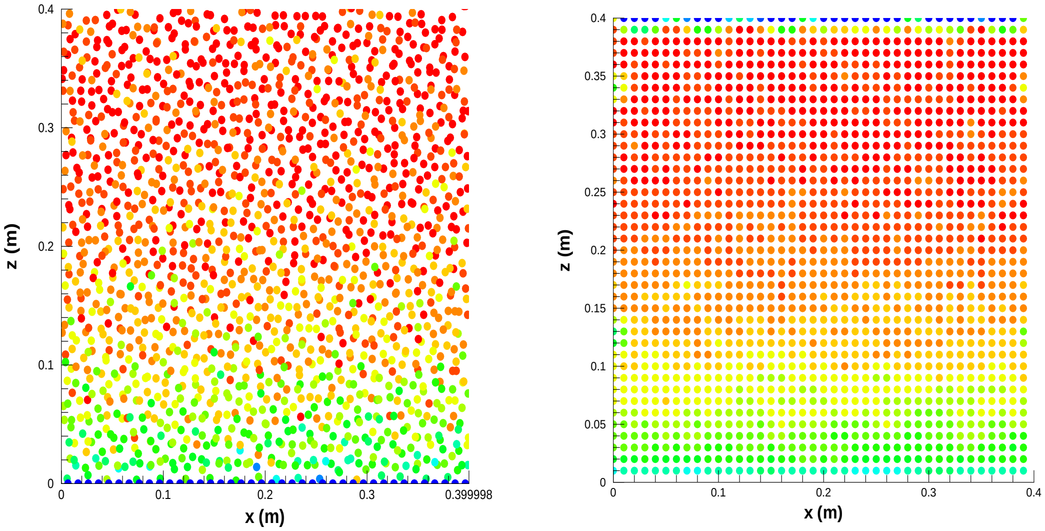

Figure 6.20: Filtered axial velocity (left) and corresponding interpolated (right) fields.

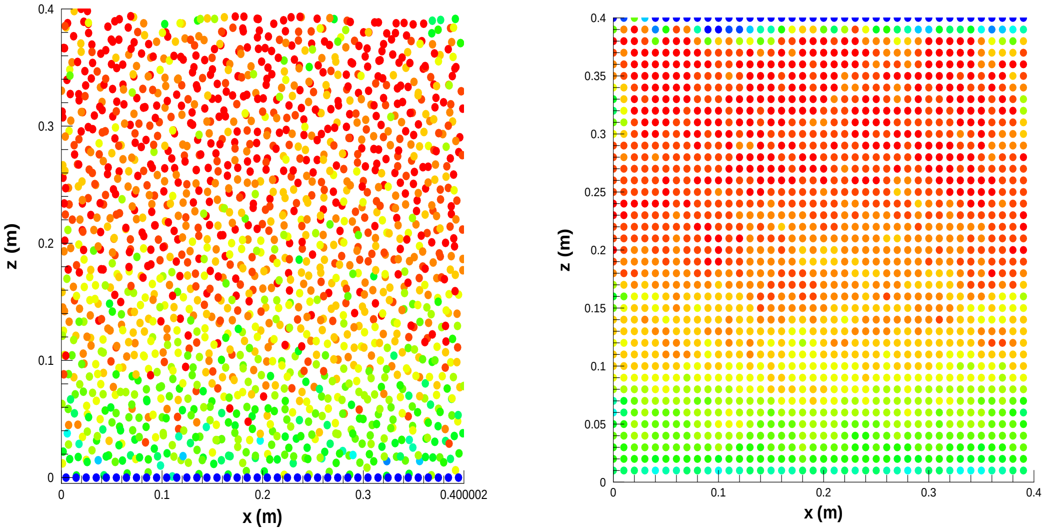

Figure 6.21: Filtered axial velocity (left) and corresponding interpolated (right) fields.

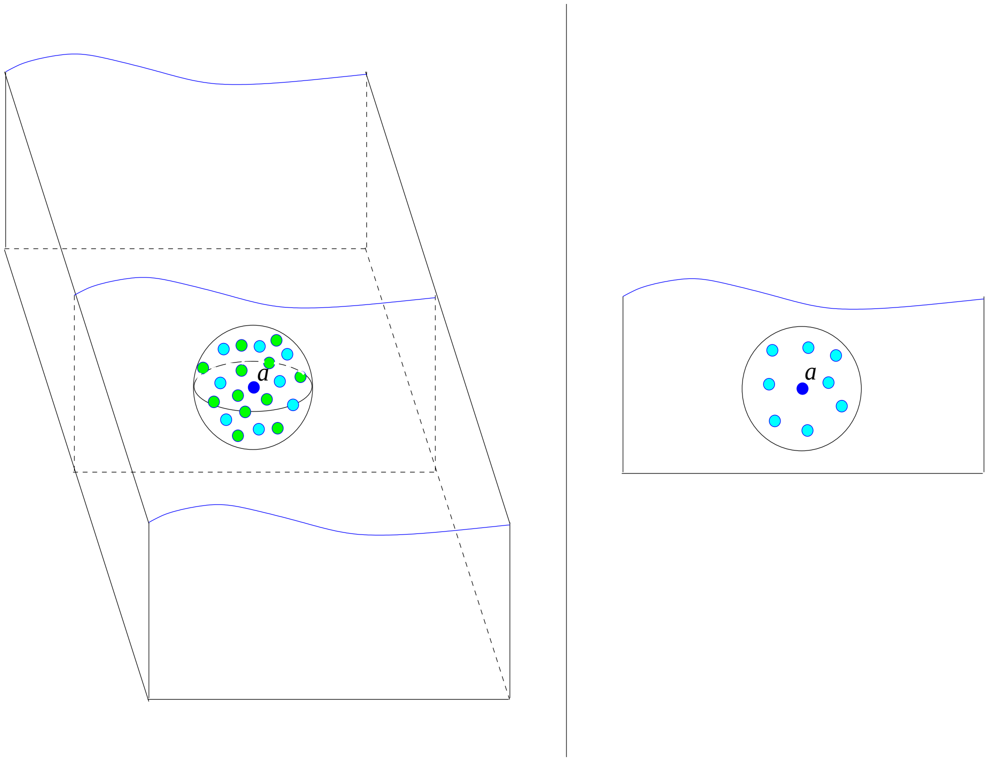

Figure 6.22: Number of neighbours in a 3D (left) and a 2D (right) approach.

Link list and asymmetrical principles

In order to efficiently select the neighbours for a particle, the "link list" method (see part 3.4.1) used in Spartacus-2D is directly adapted to Spartacus-3D. The square cells used in 2D are here replaced by cubic cells. Therefore, for a particle a located in a cubic cell, all the particles located in a-cell and in the 26 adjacent ones are considered as potential neighbours.
Once again, symmetry principles described in part 3.4.3 are used to reduce CPU and memory costs.

Code structure

The simplified algorithm structure of Spartacus-3D for a LES is described in figure 6.23.

6.4 Large Eddy Simulation of a 3D turbulent free surface channel with Spartacus-3D

The SPH Smagorinsky model implemented in Spartacus-3D is here applied to the simulation of the outer layer of a 3D turbulent free surface channel.

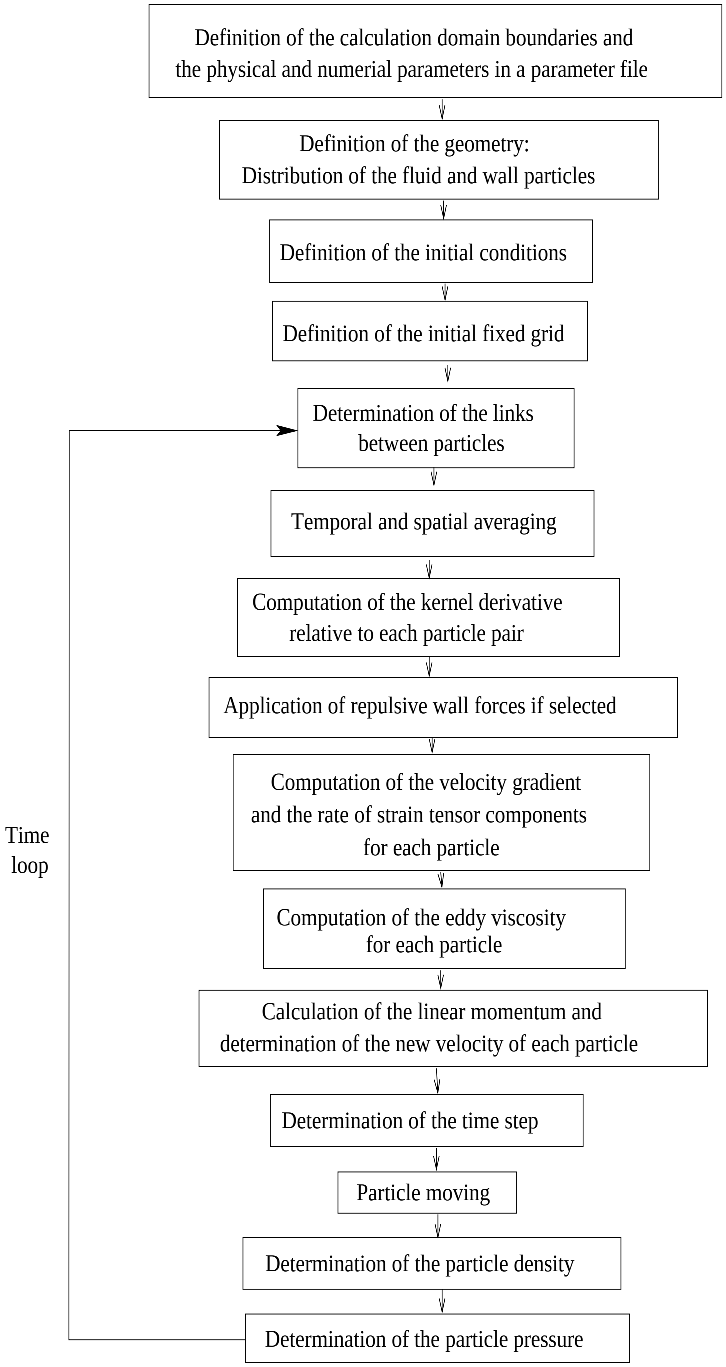
Definition of the calculation domain boundaries and the physical and numerial parameters in a parameter file
Definition of the geometry: Distribution of the fluid and wall particles
Definition of the initial conditions
Definition of the initial fixed grid
Determination of the links between particles
Temporal and spatial averaging
Computation of the kernel derivative relative to each particle pair
Application of repulsive wall forces if selected
Computation of the velocity gradient and the rate of strain tensor components for each particle
Computation of the eddy viscosity for each particle
Calculation of the linear momentum and determination of the new velocity of each particle
Determination of the time step
Particle moving
Determination of the particle density
Determination of the particle pressure
Time loop

Figure 6.23: Simplified algorithm structure of Spartacus-3D for a LES.

Fluid particles (x, z, y)	$120 \times 40 \times 20$
Edge particles	2 400
Mirror particles	7 200
Total particle number	105 600
Particle initial spacing: δr (cm)	1.

Table 6.6: Fluid, edge and mirror particle discretisation for the 3D turbulent free surface channel.

6.4.1 System modelling

Simulation conditions

The physical characteristics of the simulated flow are identical to those considered in part 6.2.1 (see table 6.2).

System discretisation and boundary conditions

The 3D channel, represented in figure 6.24, must be long and large enough in order to represent the large structures of the flow [83]. Periodic conditions are applied in both axial and spanwise (respectively x and y) directions and the system discretisation is described in table 6.6. The concept of edge particles is once again introduced and the boundary conditions are identical to those described in part 6.2.1. Nevertheless, the boundary velocity is imposed to the theoretical value in order to spare computing time, and only the case of a smooth bed is herein considered.

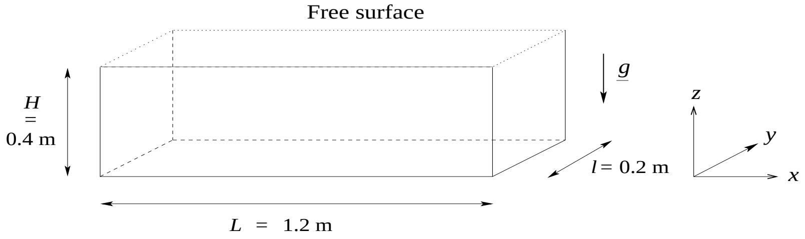

Figure 6.24: 3D free surface channel scheme.

Initial conditions

When achieving a LES with an Eulerian code, it is quite common to start from an averaged velocity field [74] corresponding to an analytical solution or obtained with a statistical precursor calculation. Moreover, in order to artificially create some turbulence [83], some random noise can be added at the beginning of the calculation to the averaged field. In the case considered here, the initial velocity field is given by the log-law (6.23) (see the left picture of figure 6.25). However, it seems that the introduction of random noise in SPH calculations is not necessary

Order of the spline kernel	3
$h/\delta r$	1.
c_0 $\left(m.s^{-1}\right)$	50

Table 6.7: Main SPH numerical parameters for the 3D tubulent free surface channel.

to create artificial turbulence. Indeed, the Lagrangian motion seems to induce some seeding perturbations by itself.

SPH equations and numerical parameters

As for the SPH mixing length model, the filtered velocity gradients for edge and fluid particles are determined with equation (5.73). The SPH Smagorinsky model (5.72) with a constant $C_S = 0.1$ is then used to compute the eddy viscosity of these particles. The filtered momentum equation for particle a is thus identical, from a formalistic point of view, to the averaged momentum equation (6.38), i.e.

$$\frac{d\underline{\overline{u}}_a}{dt} = -\sum_b m_b \left(\frac{\overline{p}_a + \overline{p}_b}{\rho_a \rho_b} - 8 \frac{\nu_{T,a} + \nu_{T,b}}{\rho_a + \rho_b} \frac{\underline{\overline{u}}_{ab} \cdot \underline{r}_{ab}}{r_{ab}^2 + \eta^2} \right) \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} + \underline{g} + \underline{F}_a^e \tag{6.48}$$

where $\underline{F}_a^e$ corresponds to the external driving force applied to each fluid particle. As previously done, this force is updated at each time step in order to impose the correct mass flow rate. All other equations correspond to those implemented in Spartacus-3D (see part6.3.1). The main SPH numerical parameters for this test case are presented in table 6.7. The low order spline kernel and the value of the parameter $h/\delta r$ were chosen in order to reduce the number of neighbours for each particle, sparing hence the computing time. In this case, 80 neighbours are considered.

6.4.2 Simulation results

Qualitative instantaneous considerations

At variance with the previous mixing length model, the SPH Smagorinsky model does not keep the particles organised, as shown in figures 6.25 (right) and 6.26. Contrary to classical Eulerian codes, it hence reveals that the particle motion induces some fluctuations, without considering any artificial noise. In order to have better visualisations, the particle axial velocity field of figure 6.25 is interpolated on a grid [2] by a postprocessing operation (see figure 6.27). One can notice that the slices extracted from this field and represented in figures 6.27 and 6.28 are typically turbulent, with velocity fluctuations in all directions [82]. As shown in figure 6.29, the fluctuations keep the instantaneous profiles relatively close to the average one.

[2] This grid must not be confused with the one used for the averaging process introduced in part 6.3.2.

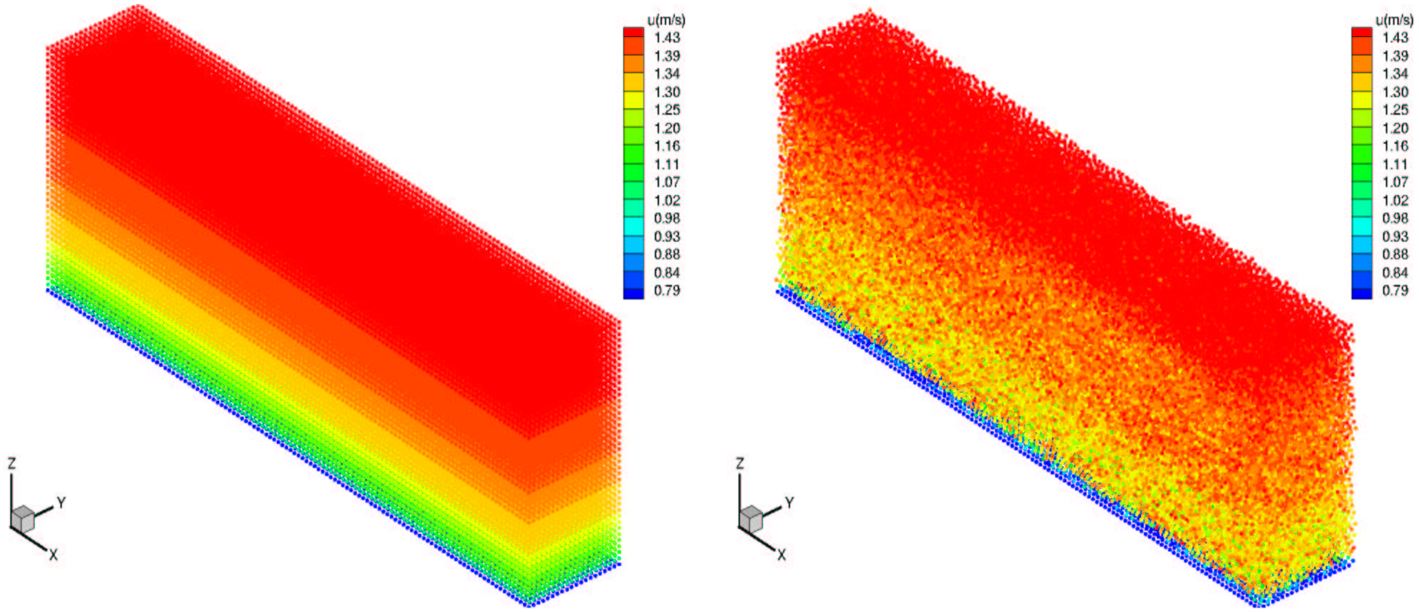

Figure 6.25: Initial axial velocity field (left) and instantaneous axial velocity field after convergence (right).

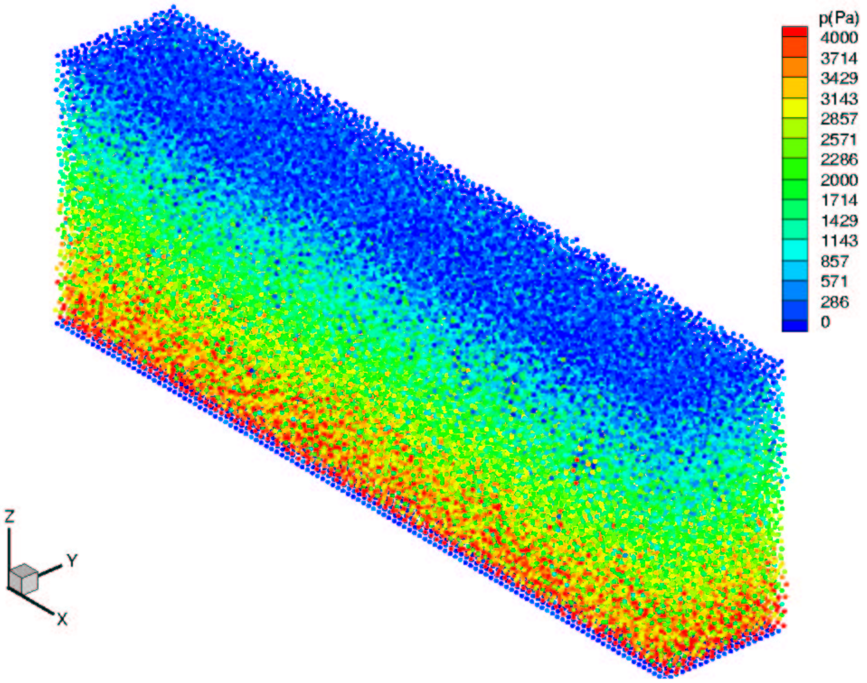

Figure 6.26: Instantaneous pressure field after convergence.

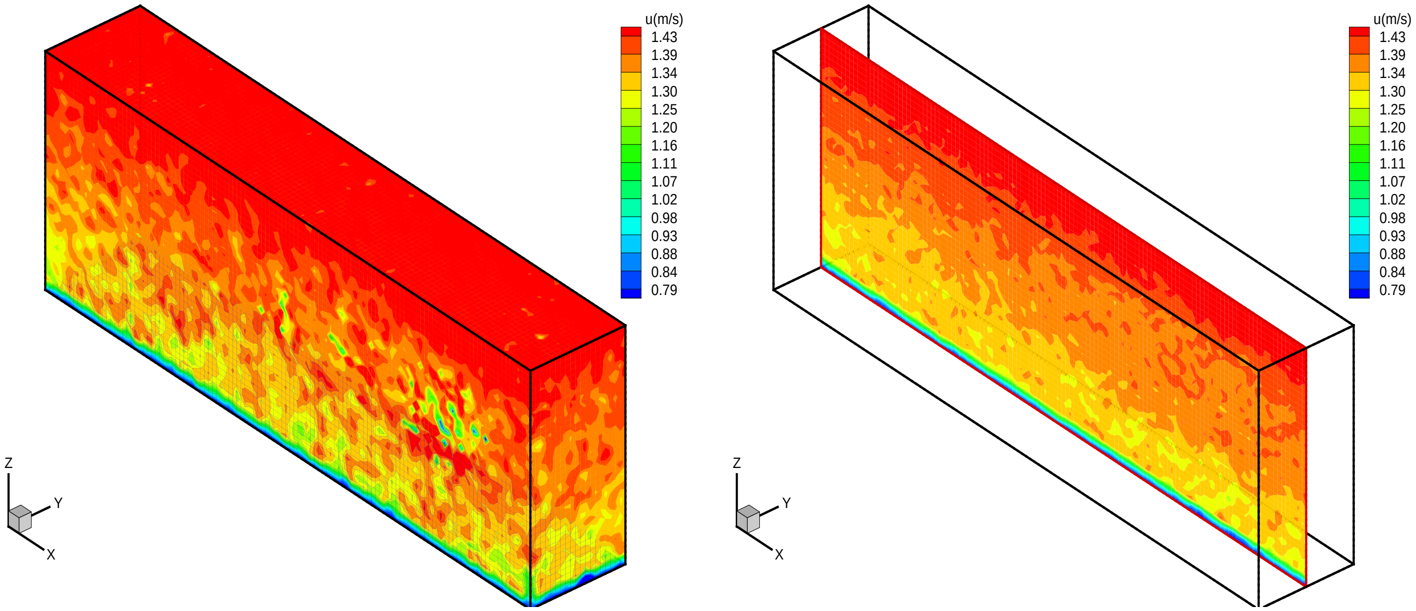

Figure 6.27: Instantaneous axial velocity field after convergence interpolated on a fixed grid (left) and a (x, z) slice in the centerline of the channel (right).

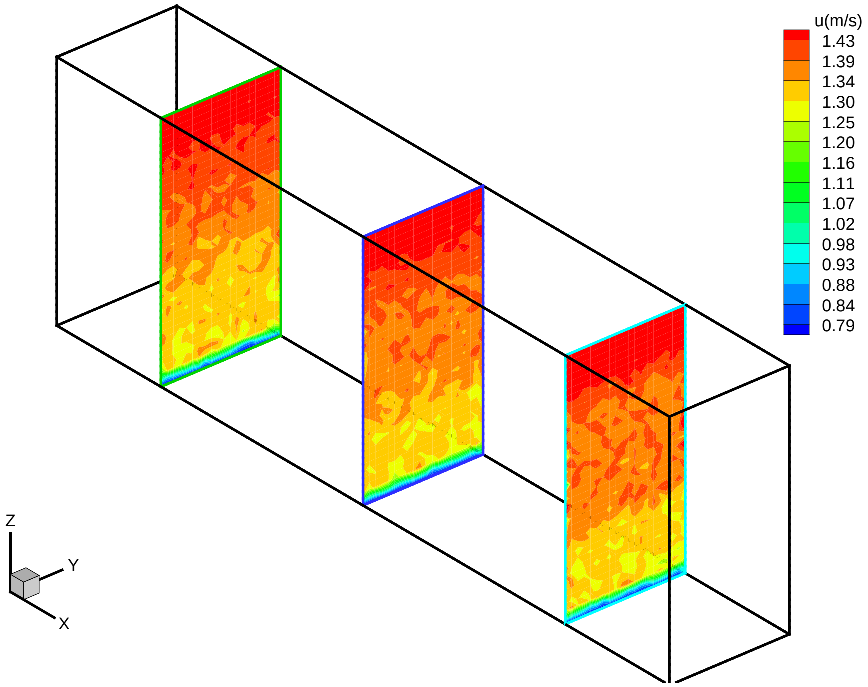

Figure 6.28: Slices (y, z) extracted from the interpolated axial velocity field of figure 6.27.

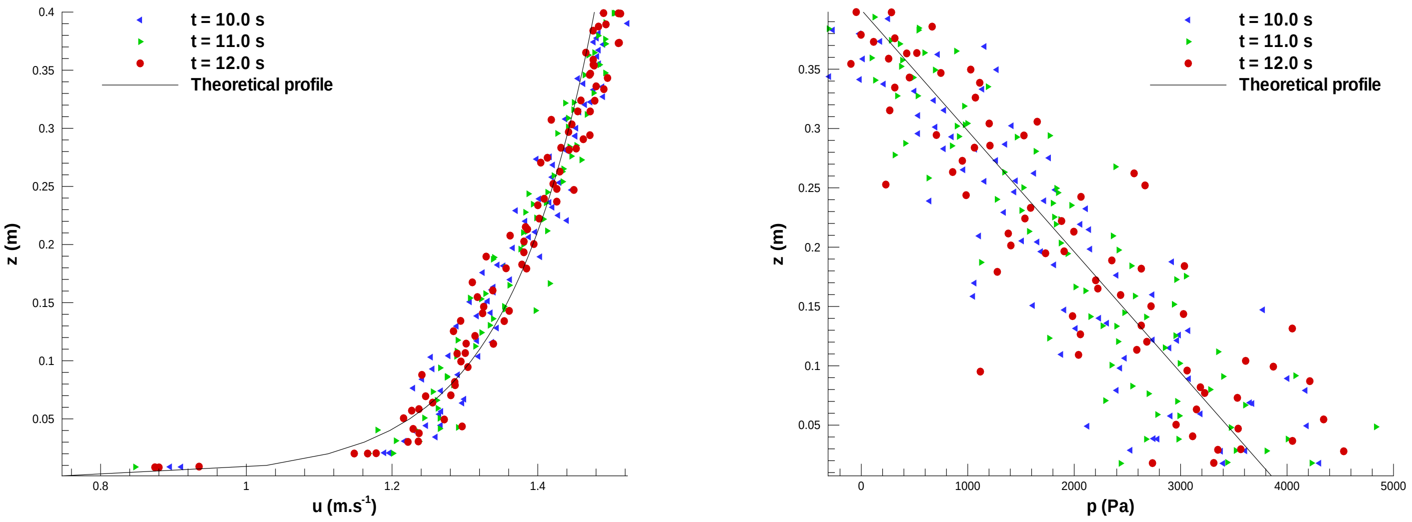

Figure 6.29: Instantaneous axial velocity (left) and pressure (right) profiles in the middle of the channel.

Averaged velocities and pressure

The averaged values for this simulation are obtained according to the process described in part 6.3.2. Since the flow is periodic, the fixed grid corresponds to the initial particle distribution (see the left picture of figure 6.25). The averaging procedure begins when a particle has been transported on a distance equal to 9 times the channel length, after which the turbulence can be considered as fully developed. In order to obtain smooth statistical values, 100 fields are considered and spatial averagings according to x and y directions are also applied. The right picture of figure 6.30 firstly proves the efficiency of the averaging process: the pressure fluctuations depicted on the right picture of figure 6.29 are completely smoothed out after averaging. Moreover, the averaged pressure profile is consistent with the theoretical one. Even if particles are characterised by 3D motions, the left picture of figure 6.30 firstly reveals that the averaged vertical and spanwise velocities are zero. It also reveals that the averaged axial velocity profile is quite close to the theoretical one. Close to the wall, the very sharp velocity gradient shows that there is insufficient mixing probably due to depleted resolved eddies. In Eulerian LES, a similar behaviour appears where the grid resolution is too coarse. This is resolved by local mesh refinement whereas in the present LES, initial particle spacing is constant. A kernel characterised by an adaptive radius could then be used.

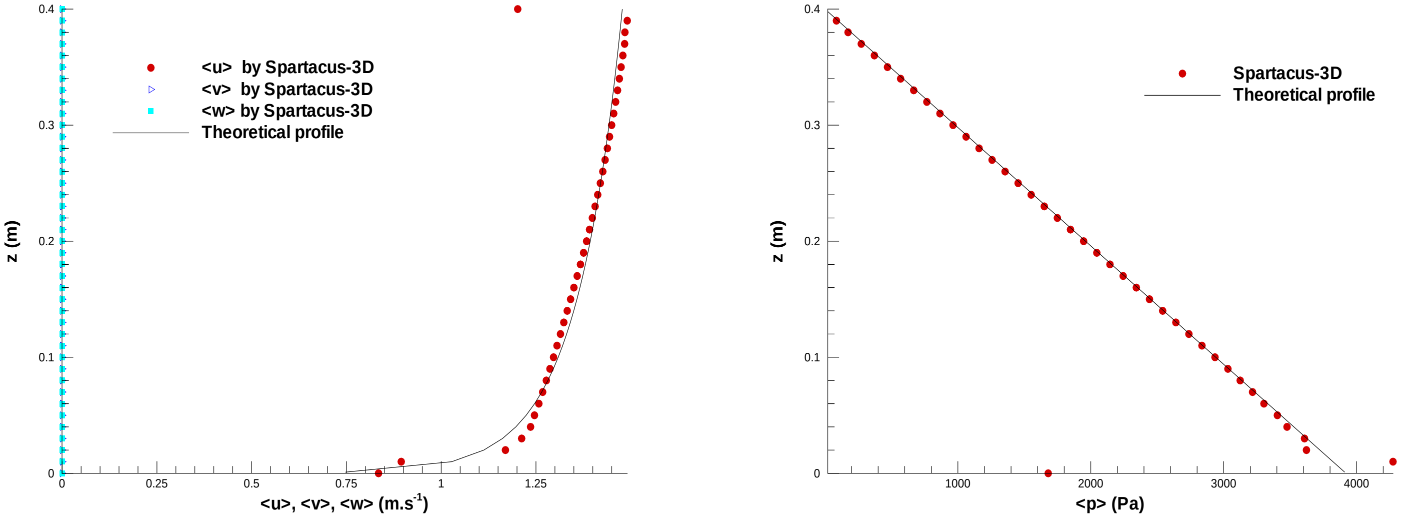

Figure 6.30: Averaged velocity (left) and pressure (right) profiles.

Turbulent intensity

In order to quantify the turbulent intensity of a flow, rms-values based on velocity fluctuations and defined by

$$u_{i,rms} = \sqrt{\langle u_i'^2 \rangle} \tag{6.49}$$

are often considered. u_i' denotes the fluctuation velocity in i-direction and $\langle . \rangle$ corresponds to the averaging operator. These values characterise the turbulent intensity in i-direction. They can be determined by postprocessing the results if enough instantaneous files (at least 50 for the considered flow) are stored during the calculation. A second way is to consider the relation

$$u_{i,rms} = \sqrt{\langle u_i^2 \rangle - \langle u_i \rangle^2} \tag{6.50}$$

which avoids the storage of many instantaneous fields but requires the time averaging of u_i^2. The second method is here implemented in Spartacus-3D. Figure 6.31 reveals that the axial (respectively vertical) fluctuations are overestimated (resp. underestimated) compared to the experimental data obtained by Nezu [66]. The short width of the channel could be responsible for these discrepancies. Indeed, a channel four times wider is usually considered with classical Eulerian codes. Due to numerical and computational limitations, Spartacus-3D can not currently handle such channels which will involve at least 400 000 particles. Moreover, the axial turbulent intensity overestimation could also be linked to the pressure estimation. Due to the state equation, the pressure gradient term may introduce extra noise when two particles are getting too close to each other. A fully incompressible method (see part 2.4.3) should then be used to avoid this numerical phenomenon. Indeed, pressure fluctuations are known to enforce return to isotropy by transfering energy from u^2 to v^2 which would reduce the discrepancy.

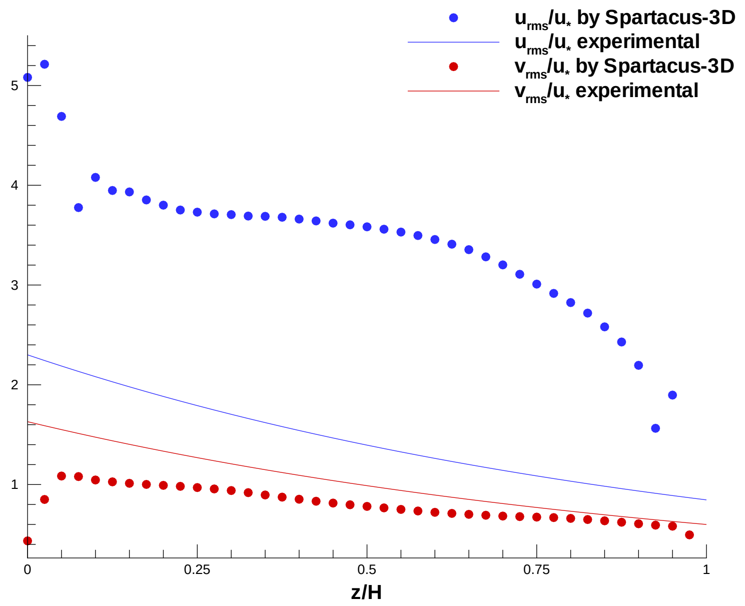

Figure 6.31: rms values of Spartacus-3D (•) compared to experimental values (——) [66].

6.4.3 Partial conclusion

Contrary to classical Eulerian codes, Spartacus-3D does not require artificial noise to generate turbulent fluctuations. The instantaneous fields after convergence are typically turbulent, with velocity fluctuations in all directions. The averaging process introduced in the SPH code is quite accurate as well and gives smooth averaged profiles. The average velocity and pressure fields obtained with Spartacus-3D are on the whole satisfactory. However, more research has to be achieved to correctly reproduce turbulent intensities and the near wall features.

6.5 Large Eddy Simulation of a 3D dam breaking with Spartacus-3D

A first attempt to apply the SPH Smagorinsky model implemented in Spartacus-3D to a complex case is presented herein: a 3D dam breaking problem, subject to gravity, is considered.

H_w	0.9
H_f	0.3
L_w	0.9
L_f	0.3
l_w	0.6
l_f	0.6

Table 6.8: Dimensions (in meters) of the dam breaking system.

6.5.1 System modelling

System discretisation

The system, periodic in the spanwise direction y, is represented in figure 6.32 and its dimensions are defined in table 6.8. The system discretisation is described in table 6.9.

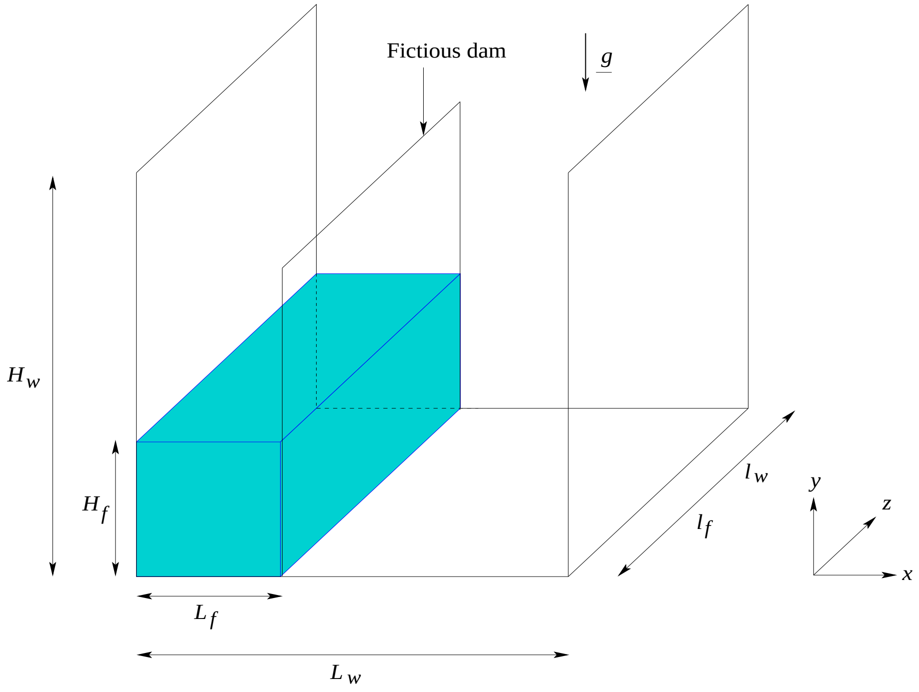

Figure 6.32: 3D dam breaking scheme.

Boundary and initial conditions

For reasons of convenience, wall and mirror particles are characterised by a zero velocity. In order to ensure the impermeability of the walls, the density of wall and mirror particles evolve at each time step according to the symmetry process described in part 2.7.2. The particles are initially distributed on a cartesian lattice, as shown the top left picture of figure 6.34. Due to the rapid dynamic characteristic of the problem, achieving a settling phase before the

Fluid particles (x, z, y)	$30 \times 30 \times 60$
Wall particles	16 680
Mirror particles	50 040
Total particle number	120 720
Particle initial spacing: δr (cm)	1.

Table 6.9: Fluid, wall and mirror particle discretisation for the 3D dam breaking.

Order of the spline kernel	3
$h/\delta r$	1.
$c_0 \ (m.s^{-1})$	20

Table 6.10: Main SPH numerical parameters for the 3D dam breaking.

beginning of the calculation is useless. At $t = 0$, the fictitious dam is suddenly removed and one focuses on the instantaneous fluid motion.

SPH equations and numerical parameters

All the SPH equations considered here are identical to those described for the previous channel case. However, the filtered momentum equation does not exhibit the external force $\underline{F}_a^e$, since the fluid is set in motion by gravity effects. The Smagorinsky constant is still equal to 0.1 and the main SPH numerical parameters are presented in table 6.10.

6.5.2 Setting up the experiment

In order to check the consistency of the simulation results, a basic experiment has been achieved at TU/Delft. The channel represented on figure 6.33 has been used and the sluice gate has been placed at a position consistent with the ratio H_f/L_f. It is removed with a manual pulley system, as represented in figure 6.33. A digital camera Sony DSC-P5 of 25 Hz has been used for the visualisation, and frames of size 320 × 240 are considered for the comparisons.

6.5.3 Simulation results

Figure 6.34 to 6.43 represent the fluid motion simulated by Spartacus-3D and compared to the experimental data, at similar non-dimensional times defined by

$$T_d = \frac{t_i}{\sqrt{\frac{2g}{H_i}}} \tag{6.51}$$

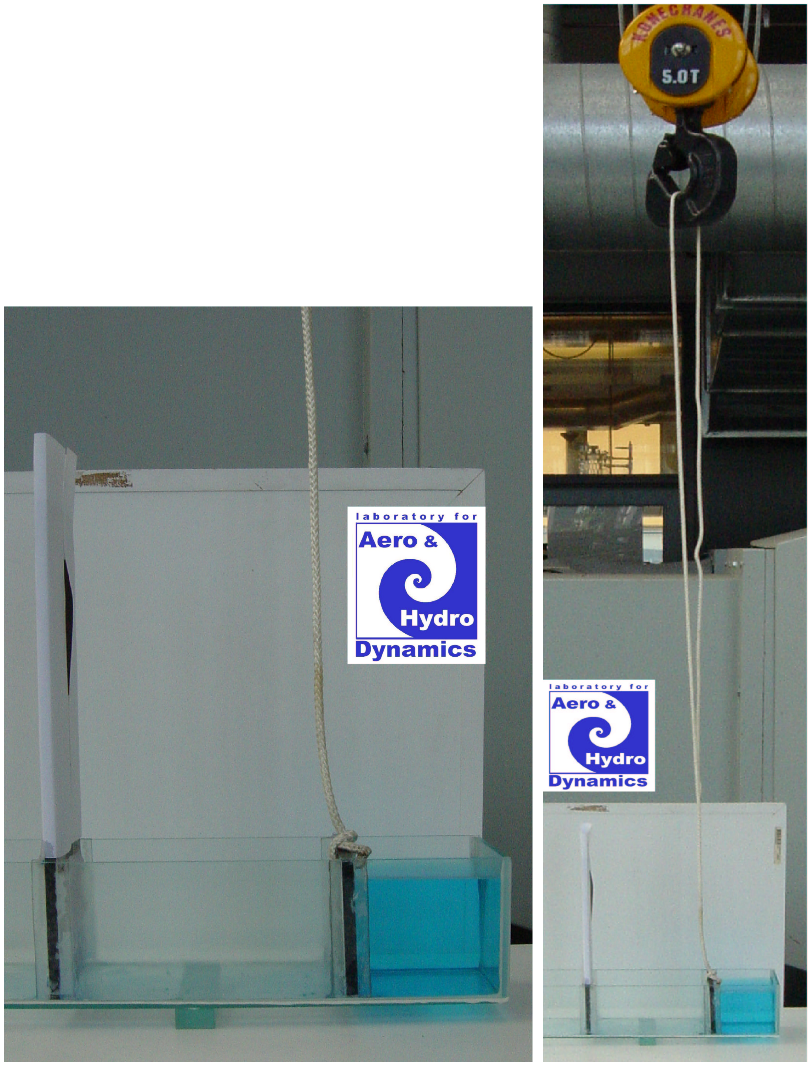

Figure 6.33: Dam breaking experiment set up (left) and manual pulley system to “instantaneously” remove the sluice gate (right).

where t_i denotes the physical time of the simulation (or of the experiment), H_i the initial height of the fluid considered by Spartacus-3D (or in the experiment) and g the gravity.

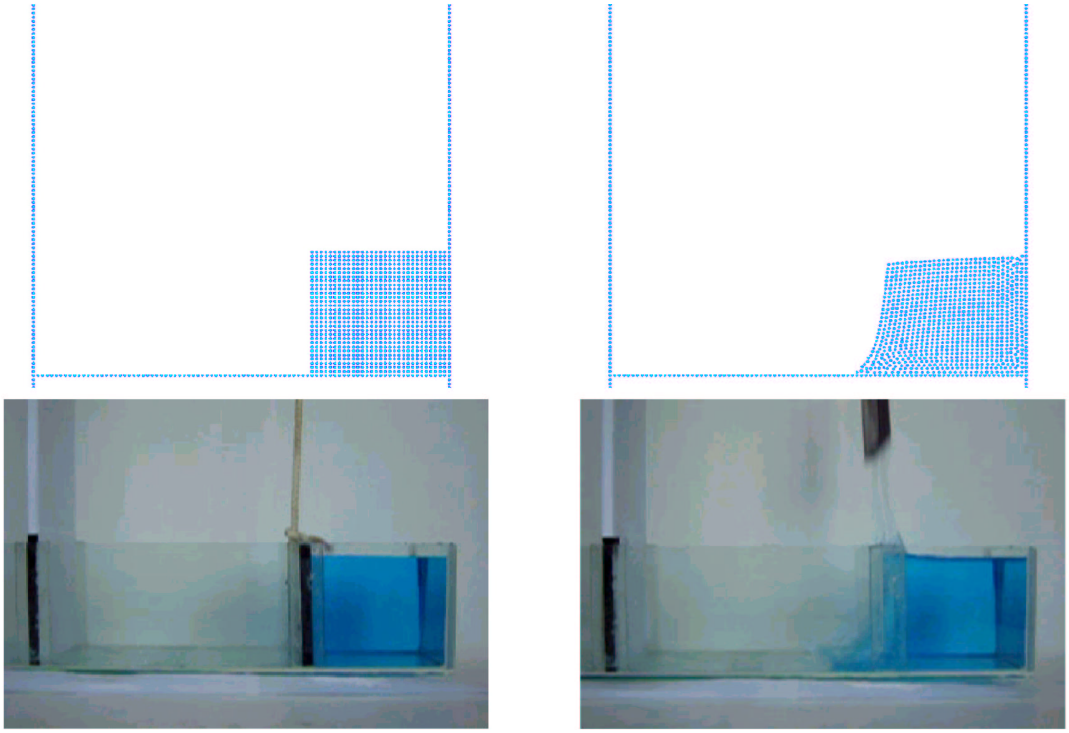

Figure 6.34: Spartacus-3D results (top) compared to experimental results (bottom) at $T_d = 0$ (left) and around $T_d = 1.0.10^{-2}$ (right).

From a qualitative point of view, the results of Spartacus-3D are on the whole quite satisfactory. The wave breaking depicted in figure 6.38 is very similar to the experimental one. Moreover, the reflection wave motion (see figure 6.42) and the fluid behaviour near walls (see figures 6.41 and 6.43) are also accurate. However, the elastic behaviour of the wave simulated by Spartacus-3D in figure 6.39 is not realistic and can be linked to the poor pressure estimation or to a dissipation underestimation.

6.5.4 Partial conclusion

In order to accurately validate Spartacus-3D for this test case, experimental measurements with PIV for instance could be achieved. However, the first results presented here are promising and more investigation will be done in the future.

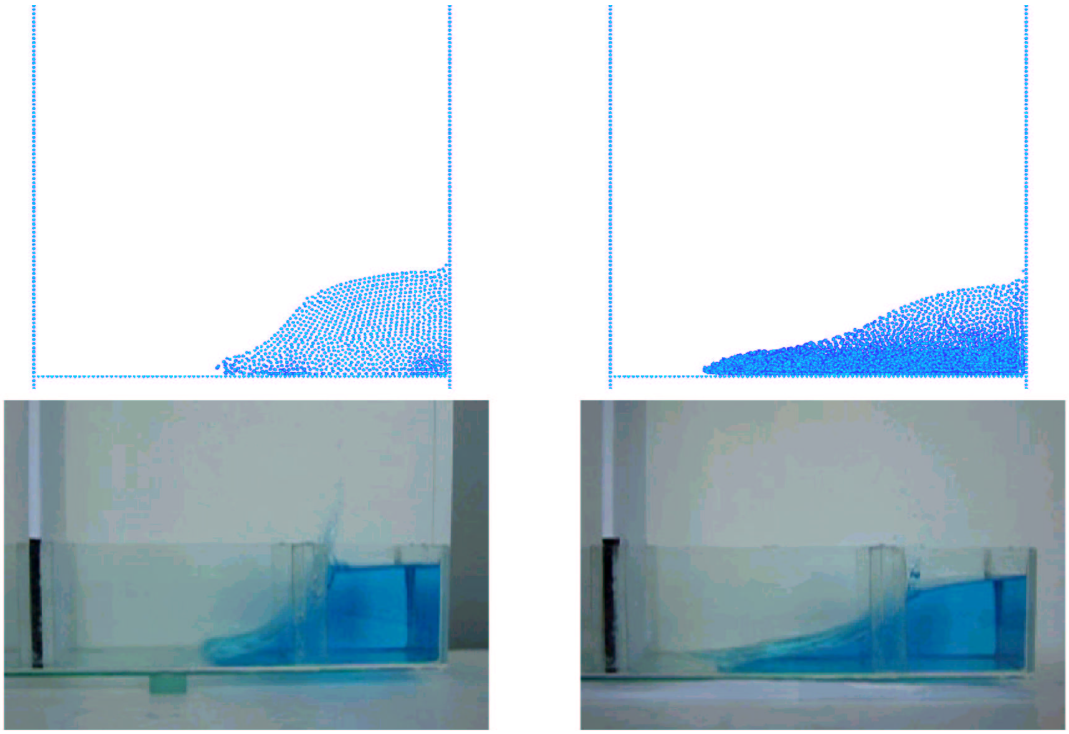

Figure 6.35: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 2.0.10^{-2}$ (left) and $T_d = 3.0.10^{-2}$ (right).

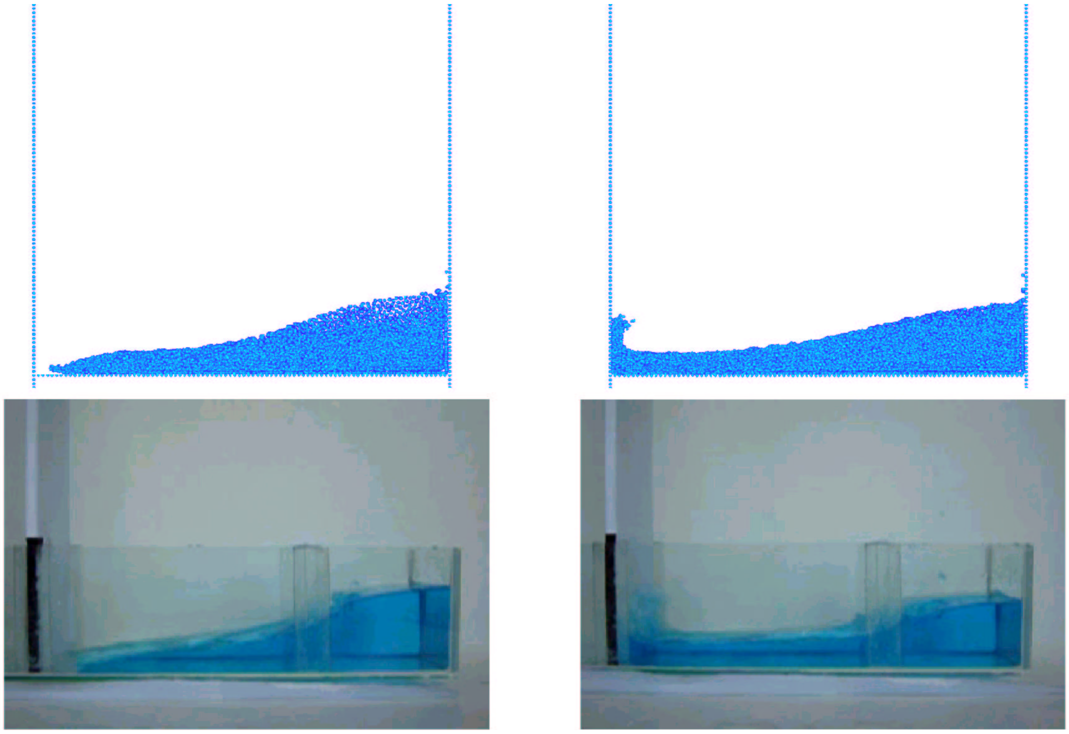

Figure 6.36: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 3.7.10^{-2}$ (left) and $T_d = 4.5.10^{-2}$ (right).

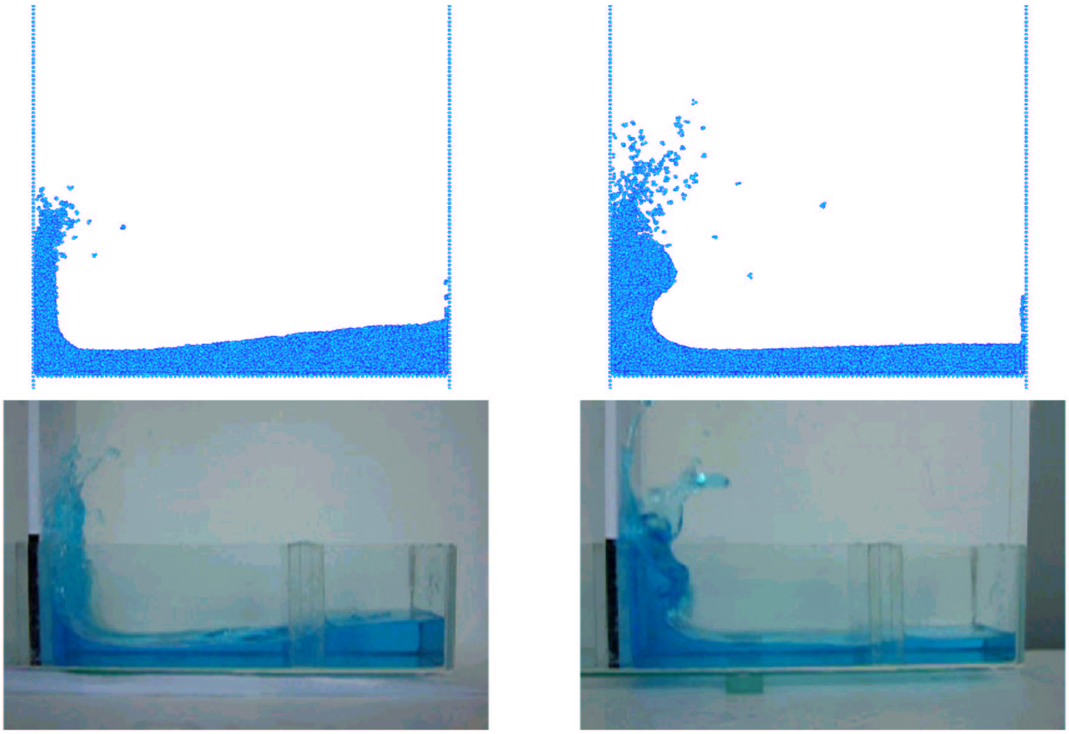

Figure 6.37: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 6.0.10^{-2}$ (left) and $T_d = 8.7.10^{-2}$ (right).

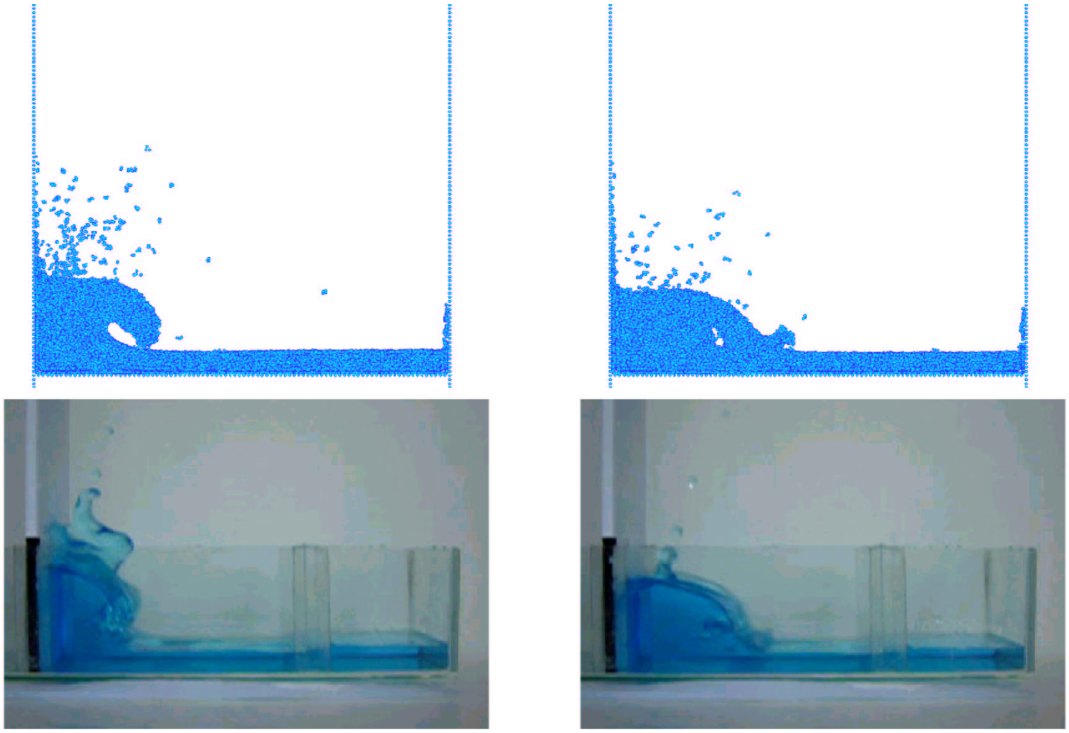

Figure 6.38: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 1.0.10^{-1}$ (left) and $T_d = 1.1.10^{-1}$ (right).

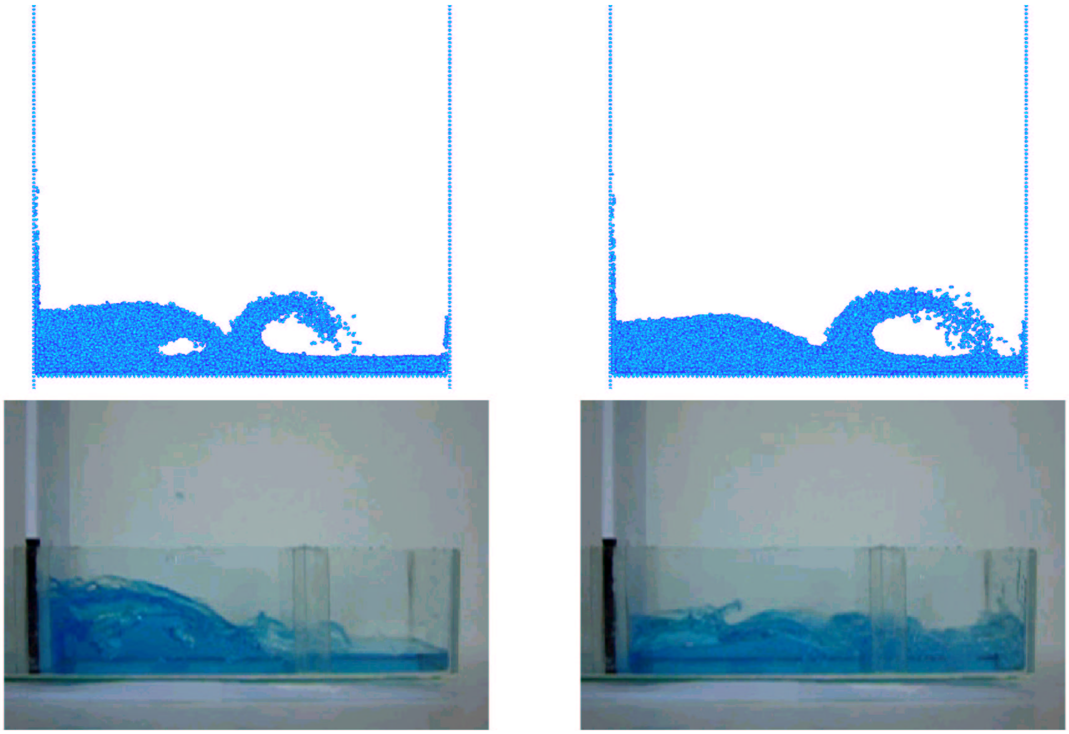

Figure 6.39: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 1.3.10^{-1}$ (left) and $T_d = 1.4.10^{-1}$ (right).

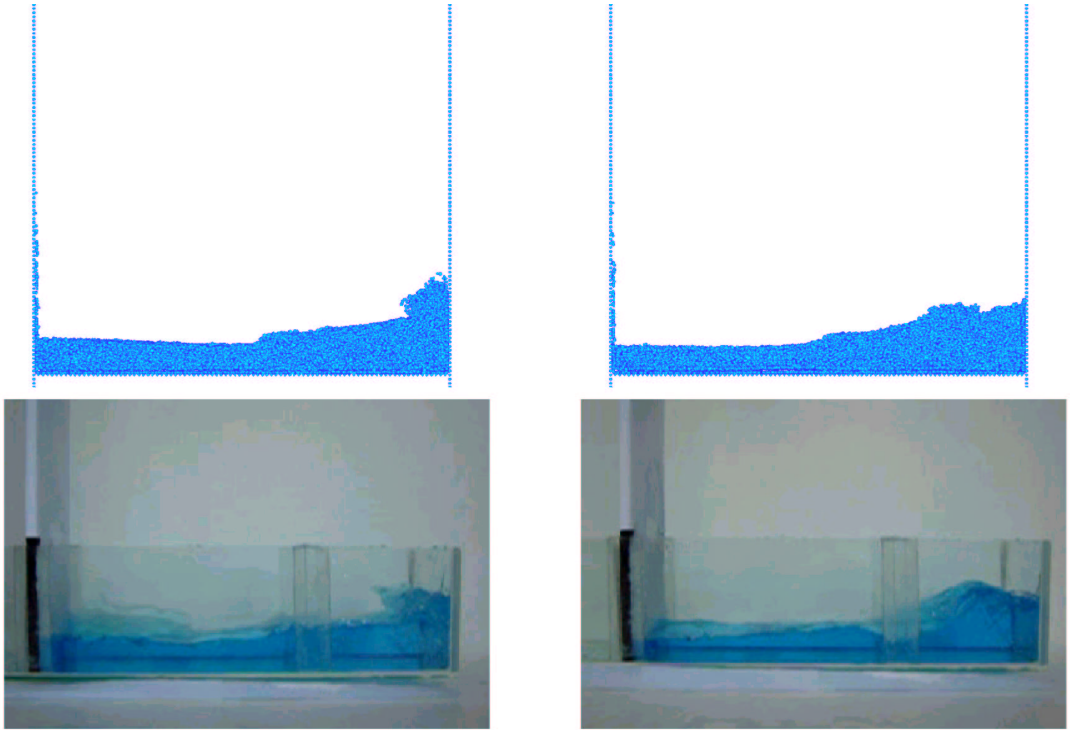

Figure 6.40: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 1.8.10^{-1}$ (left) and $T_d = 2.1.10^{-1}$ (right).

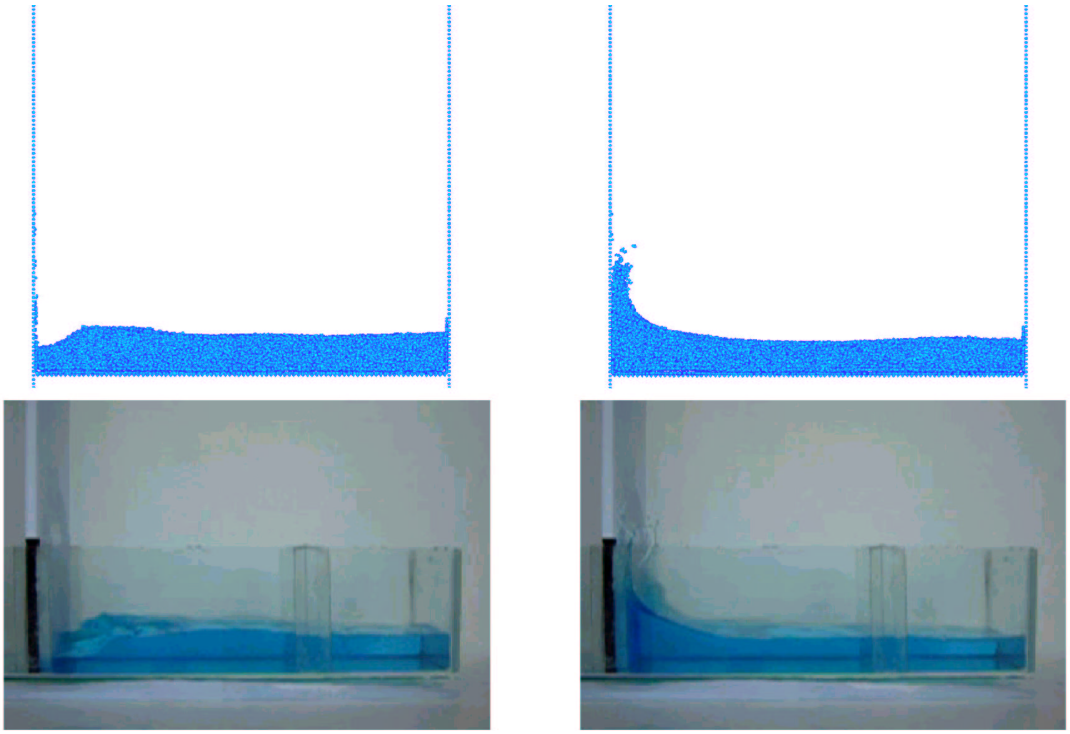

Figure 6.41: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 2.7.10^{-1}$ (left) and $T_d = 3.0.10^{-1}$ (right).

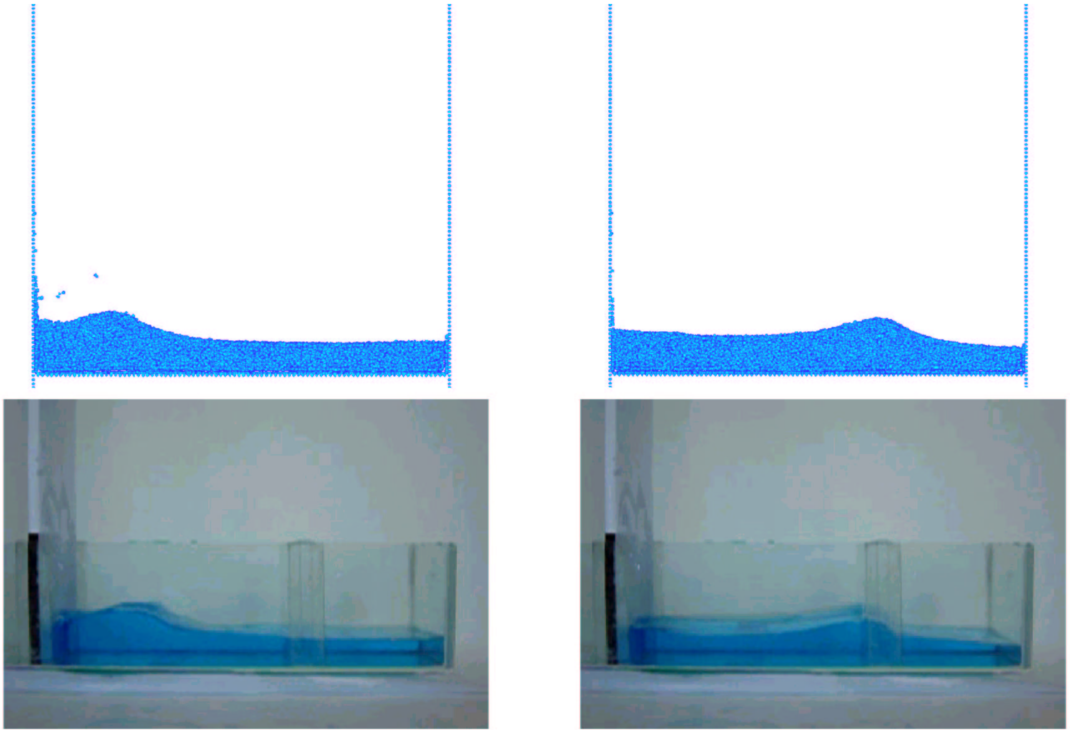

Figure 6.42: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 3.2.10^{-1}$ (left) and $T_d = 3.7.10^{-1}$ (right).

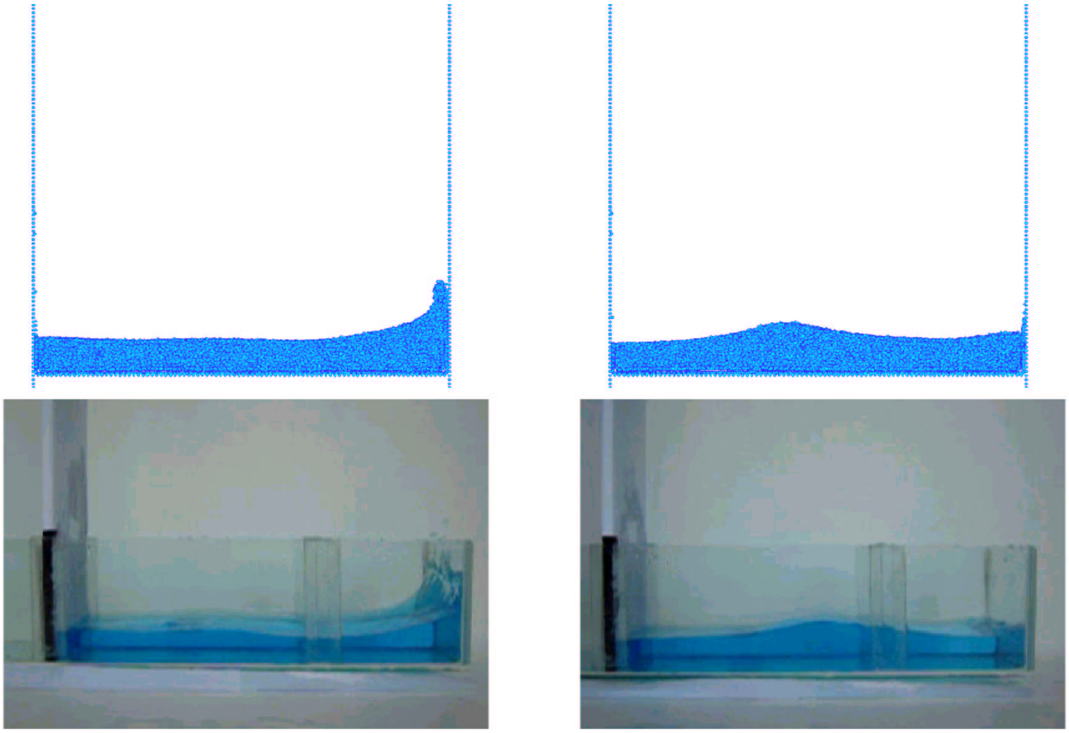

Figure 6.43: Spartacus-3D results (top) compared to experimental results (bottom) around $T_d = 4.2.10^{-1}$ (left) and $T_d = 4.8.10^{-1}$ (right).

Chapter 7

Conclusion and future work

Some laminar validations of SPH were firstly performed in this work. The investigation of a settling phase in a 2D free surface channel illustrated the problem of tensile instabilities; the introduction of a damping term successfully removed these perturbations and an accurate hydrostatic field was obtained. The study of a laminar flow in the same channel emphasized the importance of viscous effect modelling and boundary conditions: in order to reach a satisfactory axial velocity profile, viscous effects were modelled through tangential forces. Moreover, no-slip boundary conditions at the wall require to impose an asymmetric velocity to the mirror particles. The ability of SPH to reproduce laminar recirculation zones was also revealed through a 2D hill flow simulation. For this test case, an averaging procedure was developed to smooth the slight unsteadiness due to the Lagrangian characteristic of SPH and the velocity fields were then close to those obtained with an Eulerian code. The detachment and reattachment point predictions were accurate as well. The simulation of a laminar flow in a 2D backward facing step geometry proved the limitation of the classical nearly incompressible assumption used in most SPH codes. To avoid particle clumping, a high speed of sound was considered, leading to a relevant reattachment point prediction. However, the computing times were higher and the velocity profiles still revealed some slight discrepancies compared to those obtained with classical Eulerian codes.

In order to simulate turbulent flows, a SPH mixing length model was developed and successfully applied to a 2D free surface channel. For smooth and fully rough beds, the velocity profiles were very close to the theoretical ones. The drawback of this model was an overestimation of the eddy viscosity, due to an overprediction of the velocity gradient. Nevertheless, this problem is overcome by considering more complex turbulent models, such as one or two equation models.

The last part of this work presented a first attempt to adapt Large Eddy Simulation concept to SPH. A simple Smagorinsky model was applied to a 3D turbulent free surface channel and gave satisfactory averaged profiles. However, due to the pressure calculation, the turbulent intensity was not perfectly computed. The same model was successfully applied to a 3D dam breaking: the comparison of the computed free surface motion with a real experiment was quite realistic.

Since this work revealed the limitation of the nearly incompressible assumption, the development of a fully incompressible method would be of great interest: the calculation stability would probably be improved and the time step would likely to be reduced. Since the 3D calculations also emphasized the huge calculation time required by 3D-SPH, a parallelisation of any SPH code seems to be useful. Finally, the implementation of new temporal integration schemes should increase the time step and thus reduce the calculation times.

Appendix A

Viscous effects modelled by Morris

The aim of this annex is to proove that viscous effects modelled by Morris' formula (2.77) are consistent with the term $\nu \Delta \underline{u}$ of the momentum equation (2.50). Such a proof was also given in [93] for Monaghan's formulation (2.78). According to Morris, the viscous force relative to particle a can be written:

$$\underline{F}_a^{\nu} = \sum_b \frac{m_b \left(\mu_a + \mu_b\right) \underline{r}_{ab} . \dot{w}_h \left(r_{ab}\right) \underline{\epsilon}_{ab}}{\rho_a \rho_b \left(r_{ab}^2 + \eta^2\right)} \underline{u}_{ab} \tag{A.1}$$

where $\underline{\epsilon}_{ab} = \underline{r}_{ab} / r_{ab}$. Assuming constant viscosity and density and neglecting η^2, equation (A.1) gives:

$$\underline{F}_a^{\nu} = 2\nu \sum_b \frac{m_b}{\rho_b} \frac{\dot{w}_h \left(r_{ab}\right)}{r_{ab}} \underline{u}_{ab} \tag{A.2}$$

A second order Taylor development of the velocity $\underline{u}_{ab}$ written with Einstein's conventions gives:

$$\underline{F}_a^{\nu} = 2\nu \sum_b \frac{m_b}{\rho_b} \frac{\dot{w}_h \left(r_{ab}\right)}{r_{ab}} \left[\left. \frac{\partial u_i}{\partial x_j} \right|_a r_{ab}^j \underline{e}_i - \frac{1}{2} \left. \frac{\partial^2 u_i}{\partial x_j \partial x_k} \right|_a r_{ab}^j r_{ab}^k \underline{e}_i \right] \tag{A.3}$$

where $r_{ab}^j = \underline{r}_{ab} . \underline{e}_j$, $\underline{e}_j$ corresponding to the unit vector relative to j-direction. With $\epsilon_{ab}^j = \underline{\epsilon}_{ab} . \underline{e}_j$, equation (A.3) is written:

$$F_a^{\nu} = 2\nu \sum_b \frac{m_b}{\rho_b} \dot{w}_h \left(r_{ab}\right) \left. \frac{\partial u_i}{\partial x_j} \right|_a \epsilon_{ab}^j \underline{e}_i - \nu \sum_b \frac{m_b}{\rho_b} \dot{w}_h \left(r_{ab}\right) r_{ab} \left. \frac{\partial^2 u_i}{\partial x_j \partial x_k} \right|_a \epsilon_{ab}^j \epsilon_{ab}^k \underline{e}_i \tag{A.4}$$

In continuum formalism, equation (A.4) is written

$$F_a^\nu = 2\nu \int_\Omega \dot{w}_h(r) \left.\frac{\partial u_i}{\partial x_j}\right|_a \epsilon^j \underline{e}_i d\Omega - \nu \int_\Omega \dot{w}_h(r) r \left.\frac{\partial^2 u_i}{\partial x_j \partial x_k}\right|_a \epsilon^j \epsilon^k \underline{e}_i d\Omega \tag{A.5}$$

where Ω corresponds to the influence domain of particle a. If this domain is a disc, equation (A.5) becomes

$$F_a^\nu = 2\nu \left.\frac{\partial u_i}{\partial x_j}\right|_a \underbrace{\int_{r=0}^{h_t} \dot{w}_h(r) r dr}_{I} \underbrace{\int_{\theta=0}^{2\pi} \epsilon^j d\theta \underline{e}_i}_{A_{ji}} - \nu \left.\frac{\partial^2 u_i}{\partial x_j \partial x_k}\right|_a \underbrace{\int_{r=0}^{h_t} \dot{w}_h(r) r^2 dr}_{J} \underbrace{\int_{\theta=0}^{2\pi} \epsilon^j \epsilon^k d\theta \underline{e}_i}_{B_{jki}} \tag{A.6}$$

$$F_a^\nu = 2\nu \left.\frac{\partial u_i}{\partial x_j}\right)_a I A_{ji} - \nu \left.\frac{\partial^2 u_i}{\partial x_j \partial x_k}\right)_a J B_{jki} \tag{A.7}$$

ϵ^i denotes the i-coordinate of the unit vector oriented from particle b and pointed towards particle a. For instance, in a 2D repair (x, y) where a is the origin, $\epsilon^i = cos\theta$ if $i = x$ and $\epsilon^i = sin\theta$ if $i = y$. In equation (A.7), since the first order term is zero, only the second order term remains. The quantities B_{jki} of the second order term are hence summations such as $\int_{\theta=0}^{2\pi} cos^p\theta . sin^q\theta d\theta$ which are zero except if p and q are both even. Non zero quantities must check $B_{xx} = B_{zz} = \pi$.

An integration by parts of J gives

$$J = -2 \int_{r=0}^{h_t} r w_h(r) dr \tag{A.8}$$

Regardless of the kernel, the condition $\int_\Omega w_h(r) d\Omega = 1$ implies $J = -1/\pi$. Consequently, the viscous force applied on particle a is written:

$$\underline{F}_a^\nu = \nu \left[\left.\frac{\partial^2 u}{\partial x^2}\right|_a \underline{e}_x + \left.\frac{\partial^2 u}{\partial z^2}\right|_a \underline{e}_x + \left.\frac{\partial^2 v}{\partial x^2}\right|_a \underline{e}_z + \left.\frac{\partial^2 v}{\partial z^2}\right|_a \underline{e}_z \right] \tag{A.9}$$

Contrary to the proof relative to Monaghan's formula established in [93], it is thus not necessary to consider the continuity equation to proove that

$$\underline{F}_a^\nu = \nu \Delta \underline{u}_a \tag{A.10}$$

Appendix B

Necessary stability condition

The aim of this annex is to derive the stability criterion (2.119). This proof was achieved by Violeau in [86].
The potential energy of a static fluid is defined according to

$$E_p = \frac{p}{\rho} + gz = -\frac{T}{\rho} \tag{B.1}$$

where T denotes an equivalent stress. In SPH formalism, the potential energy of a particle a is written

$$E_{p,a} = \sum_b E_{p,b} w_h \left(r_{ab}\right) \tag{B.2}$$

The Taylor development of a function F around a point located at $\underline{r}_a$ gives

$$F = F_a + \underline{G}_a . \underline{\delta} + \frac{1}{2} \underline{\delta} \underline{\underline{H}}_a \underline{\delta} + O\left(|\delta|^2\right) \tag{B.3}$$

where

$$\underline{\delta} = \underline{r} - \underline{r}_a \tag{B.4}$$

$$\underline{G}_a = \left.\frac{\partial F}{\partial x_i}\right|_a \underline{e}_i = \left(\nabla F\right)_a \tag{B.5}$$

$$\underline{\underline{H}}_a = \left.\frac{\partial^2 F}{\partial x_i \partial x_j}\right|_a \underline{e}_i \otimes \underline{e}_j \tag{B.6}$$

It is now possible to develop (B.2):

$$\left.\frac{\partial E_{p,a}}{\partial x_i}\right|_a = \sum_b E_{p,b} \frac{\partial w_h\left(r_{ab}\right)}{\partial x^i_{ab}} = \sum_b E_{p,b} \frac{x^i_{ab}}{r_{ab}} \dot{w}_h\left(r_{ab}\right) \tag{B.7}$$

Since

$$\underline{G}_a = \left(\nabla E_{p,a}\right)_a = \sum_b E_{p,b} \nabla_a w_h\left(r_{ab}\right) \tag{B.8}$$

one obtains

$$\underline{G}_a = \sum_b E_{p,b} \dot{w}_h\left(r_{ab}\right) \underline{\epsilon}_{ab} \tag{B.9}$$

<u>*Remark:*</u> In a 1D problem, $\epsilon_{ab} = \pm 1$. Generally, we recall that $\underline{\epsilon}_{ab}$ denotes the ratio $\underline{r}_{ab}/r_{ab}$.

Moreover, the second derivative of the potential energy is

$$\left.\frac{\partial^2 E_{p,a}}{\partial x_i \partial x_j}\right)_a = \sum_b E_{p,b} \frac{\partial}{\partial x^j_{ab}} \left(\frac{x^i_{ab}}{r_{ab}} \dot{w}_h\left(r_{ab}\right)\right) \tag{B.10}$$

Besides,

$$\begin{aligned}\frac{\partial}{\partial x^j_{ab}} \left(\frac{x^i_{ab}}{r_{ab}} \dot{w}_h\left(r_{ab}\right)\right) = \frac{\dot{w}_h\left(r_{ab}\right)}{r_{ab}} \delta_{ij} - \frac{x^i_{ab} x^j_{ab}}{r^3_{ab}} \dot{w}_h\left(r_{ab}\right) +& \\ \frac{x^i_{ab} x^j_{ab}}{r^2_{ab}} \ddot{w}_h\left(r_{ab}\right)&\end{aligned} \tag{B.11}$$

where $\ddot{w}_h$ denotes the second derivative of the kernel and δ_{ij} the Kronecker's symbol. Therefore, it comes

$$\left.\frac{\partial^2 E_{p,a}}{\partial x_i \partial x_j}\right|_a = \sum_b E_{p,b} \left[\frac{x^i_{ab} x^j_{ab}}{r^2_{ab}} \ddot{w}_h\left(r_{ab}\right) + \frac{\dot{w}_h\left(r_{ab}\right)}{r_{ab}} \left(\delta_{ij} - \frac{x^i_{ab} x^j_{ab}}{r^2_{ab}}\right)\right] \tag{B.12}$$

then

$$\underline{\underline{H}}_a = \sum_b E_{p,b} \left[\ddot{w}_h\left(r_{ab}\right) . \underline{\epsilon}_{ab} \otimes \underline{\epsilon}_{ab} + \frac{\dot{w}_h\left(r_{ab}\right)}{r_{ab}} \left(\underline{\underline{I}} - \underline{\epsilon}_{ab} \otimes \underline{\epsilon}_{ab}\right)\right] \tag{B.13}$$

where $\underline{\underline{I}}$ is the identity tensor. In a 1D problem, the second term is zero. Then, if we consider a flow where the density is sensibly constant, one can deduce a necessary condition of equilibrium and stability from equations (B.9) and (B.12). Equilibrium corresponds in equation (B.3) to $\underline{G}_a = \underline{0}$ and stability is linked to the positive defined character of $\underline{\underline{H}}_a$:

- *Equilibrium*

$$\forall a, \quad \sum_{b} T_b \dot{w}_h \left(r_{ab} \right) \underline{\epsilon}_{ab} = \underline{0} \tag{B.14}$$

- *Stability*

$$\forall a, \quad \sum_{b} T_b \ddot{w}_h \left(r_{ab} \right) < 0 \tag{B.15}$$

The first equation means that each particle receives a zero resultant force. The second one corresponds to the stability condition (2.119).

Bibliography

[1] J. M. Alimi, A. Serna, C. Pastor, and G. Bernabeu. Smooth particle hydrodynamics: importance of correction terms in adaptative resolution algorithms. *J. Comp. Phys.*, 192:157–174, 2003.

[2] B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schonung. Experimental and Theoretical Investigation of a Backward-Facing Step. *J. Fluid Mech.*, 127:473–496, 1983.

[3] L. A. Barba, A. Leonard, and C. B. Allen. Vortex method with mesh-less spatial adaptation for accurate simulation of viscous, unsteady vortical flows. *To be published in International Journal For Numerical Methods in Fluids*, 2004.

[4] L. A. Barba, A. Leonard, and C. B. Allen. Numerical Investigations on the Accuracy of the Vortex Method With and Without Remeshing. In *16th Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics*, June 2003.

[5] J. Bonnet and S. Kulasegaram. A simplified approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computations*, 126:133–155, 2002.

[6] J. Bonnet and T.-SL. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamics. *Comput. Methods Appl. Mech. Engrg.*, 180:97–115, 1999.

[7] J. U. Brackbill. Particle methods. In *Proceedings of 2004 of Institute for Computational Fluid Dynamics conference, Oxford, UK*, 2004.

[8] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel. FLIP: a low dissipation, particle-in-cell method for fluid flows. *Comput. Phys. Commun.*, 48:25–38, 1988.

[9] J. U. Brackbill and H. M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculations in two dimensions. *J. Comp. Phys.*, 65:314–343, 1986.

[10] A. K. Chaniotis, D. Poulikakos, and P. Koumoutsakos. Remeshed Smoothed Particle Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows. *J. Comp. Phys.*, 182:67–90, 2002.

[11] A. J. Chorin. A numerical method for solving incompressible flow problems. *J. Comp. Phys.*, 2:12–26, 1967.

[12] P. W. Cleary and J. Ha. SPH modelling of isothermal High Pressure Die Casting. In *Proceedings of the 13th Australian Fluid Mechanics conference*, 1998.

[13] P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *J. Comp. Phys.*, 148:227–264, 1999.

[14] G. H. Cottet and P. Koumoutsakos. *Vortex Methods: theory and practice.* Cambridge University Press, 2000.

[15] G. H. Cottet, P. Koumoutsakos, and M. L. Ould Salihi. Vortex methods with spatially varying cores. *J. Comp. Phys.*, 162:164–185, 2000.

[16] K. Dolag, M. Bartelmann, and H. Lesch. SPH simulations of magnetic fields in galaxy clusters. *Astronomy and Astrophysics*, 348:351–363, 1999.

[17] J.G.M. Eggels. *Direct and Large Eddy Simulation of Turbulent Flow in a Cylindrical Pipe Geometry.* PhD thesis, TU/Delft, Aero en Hydrodynamica, 1994.

[18] M. Ellero, M. Kroger, and S. Hess. Viscoelastic flows studied by smoothed particle dynamics. *Journal of Non-Newtonian Fluid mechanics*, 105:35–51, 2002.

[19] S. Faure and D. Violeau. Développement d'un algorithme de projection exacte pour la modélisation des écoulements incompressibles par la méthode SPH. *Rapport EDF-LNHE / HP-72/2000/032/A*, 2000.

[20] J. H. Ferziger. Large eddy simulation. Turbulence course, Stanford University.

[21] O. Flebbe, S. Münzel, H. Herold, H. Riffert, and H. Ruder. Smoothed particle hydrodynamics: Physical viscosity and the simulation of accretion disks. *The Astrophysical Journal*, 431:754–760, 1994.

[22] R. W. Fox and A. T. McDonald. *Introduction to Fluid Mechanics, fourth ed.* Wiley, 1994.

[23] R. A. Gingold and J. J. Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *J. Comp. Phys.*, 46:429–453, 1982.

[24] J. P. Gray, J. J. Monaghan, and R. P. Swift. SPH elastic dynamics. *Computer methods in applied mechanics and engineering*, 190:6641–6662, 2001.

[25] V. Guimet and D. R. Laurence. A linearised turbulent production in the $k - \epsilon$ model for engineering applications. In *Proc. 5th International Symposium on Engineering Turbulence Modelling and Measurements, Mallorca, Spain*, 2002.

[26] J. Ha and P. W. Cleary. Comparison of SPH simulations of High Pressure Die Casting with the experiments and VOF simulations of Schmid & Klein. *Submitted to International Journal of Cast Metals Research.*

[27] K. Hanjalić and B. E. Launder. Contribution towards a Reynolds-stress closure for low-Reynolds-number turbulence. *J. Fluid Mech.*, 74:593–610, 1976.

[28] F. H. Harlow. Hydrodynamic problems involving large fluid distorsion. *J. Assoc. Comp. Mach.*, 4:137, 1957.

[29] F. H. Harlow. The Particle-In-Cell method for numerical solution of problems in fluid dynamics. In *Proc. of Symposia in Applied Mathematics 15*, pages 269–288, 1963.

[30] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow. *Phys. Fluids*, 8:2182–2189, 1965.

[31] J. M. Hervouet. *Hydrodynamique des écoulements à surface libre. Modélisation numérique avec la méthode des éléments finis.* PhD thesis, Presses de l'Ecole Nationale des Ponts et Chaussées, 2003.

[32] D. L. Hicks and L. M. Liebrock. SPH Hydrocodes Can Be Stabilized with Shape-Shifting. *Computers and Mathematics with Applications*, 38:1–16, 1999.

[33] H. Hotoh, T. Shibahara, and T. Sakai. Sub-Particle-Scale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering. *Advanced Methods for Computational Fluid Dynamics*, 9:339–347, 2001.

[34] R. Issa. Refroidissement des chambres de combustion de moteur-fusées. MSc report, INPG, 2001.

[35] R. Issa. Simulation of a free surface laminar flow with the Smoothed Particle Hydrodynamics method (SPH). 1st year PhD report, UMIST, 2002.

[36] R. Issa. Turbulent free surface and laminar hill flows with the particle method SPH. 2nd year PhD report, UMIST, 2003.

[37] R. Issa, E. S. Lee, D. Violeau, and D. Laurence. Incompressible separated flows simulations with the Smoothed Particle Hydrodynamics gridless method. *To appear in International Journal for Numerical Methods in Fluids*, 2004.

[38] R. Issa and D. Violeau. Simulation of a 2D free surface turbulent flow with SPH. *Rapport EDF-LNHE HP-75/2002/088/A*, 2003.

[39] R. Issa, D. Violeau, and D. Laurence. Simulation of a laminar free surface flow with SPH. *submitted to Journal of Hydraulic Research*, 2003.

[40] G. R. Johnson. Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations. *Nucl. Engrg. Des*, 150:265–274, 1994.

[41] W. P. Jones and B. E. Launder. The predictions of laminarization with a two-equation model of turbulence. *Int. J. Heat Mass Transfer*, 15:301–314, 1972.

[42] G. Lapenta and J. U. Brackbill. Control of the number of particles in fluid and MHD Particle-In-Cell methods. *Comput. Phys. Commun.*, 87:139–154, 1985.

[43] M. Lastiwka, N. Quinlan, and M. Basa. Adaptive particle distribution for Smoothed Particle Hydrodynamics. *To be published in International Journal for Numerical Methods in Fluids*, 2004.

[44] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a Reynolds-stress turbulence closure. *J. Fluid Mech.*, 68:537–566, 1975.

[45] B. E. Launder and D. B. Spalding. *Mathematical models of turbulence.* Academic Press, London, 1972.

[46] E. S. Lee. Simulating laminar backward-facing step flow with SPH. MSc report, UMIST, 2003.

[47] M. Lesieur. *Turbulence in fluids, third ed.* Kluwer, 1997.

[48] M. Lesieur and O. Métais. New trends in large-eddy simulations of turbulence. *Annu. Rev. Fluid Mech.*, 28:45–98, 1999.

[49] S. Li and W. K. Liu. Meshfree and particle methods and their applications. *Appl Mech Rev*, 55:1–34, 2002.

[50] P. Liovic, M. Rudman, and J. L. Liow. Numerical modelling of free surface flows in metallurgical vessels. In *Second International Conference on CFD in the Minerals and Process Industries, CSIRO, Melbourne, Australia*, pages 255–260, 1999.

[51] E. Y. M. Lo and S. Shao. Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Applied Ocean Research*, 24:275–286, 2002.

[52] X. L. Luo. Operator splitting computation of turbulent flow in an axisymmetric 180^0 narrowing bend using several k-epsilon models and wall functions. *International Journal of Numerical Methods in Fluids*, 22:1189–1205, 1996.

[53] X. L. Luo, A. N. Stokes, and N. G. Barton. Turbulent flow around a car body - Report of Fastflo solutions. Technical report, WUA-CFD Freiburg, 1996.

[54] O. Métais. Turbulence : modélisation et simulation numérique. Master of science turbulence course, INPG, 2001.

[55] S. Mittal. Finite element computation of the dynamics of large ram air parachutes. *AHPCRC Bulletin*, 4-2, 1994.

[56] J. J. Monaghan. Smoothed Particle Hydrodynamics. *Annu. Rev. Astron. Astrophy.*, 30:543–574, 1992.

[57] J. J. Monaghan. Simulating free surface flows with SPH. *J. Comp. Phys.*, 110:399–406, 1994.

[58] J. J. Monaghan. Simulating gravity currents with SPH I Lock gates. *Mathematics Reports and Preprints*, 95/5, 1995.

[59] J. J. Monaghan. Simulating gravity currents with SPH III Boundary forces. *Mathematics Reports and Preprints*, 95/5, 1995.

[60] J. J. Monaghan. Implicit SPH Drag and Dusty Gas Dynamics. *J. Comp. Phys.*, 138:801–820, 1997.

[61] J. J. Monaghan. SPH and Riemann Solvers. *J. Comp. Phys.*, 136:298–307, 1997.

[62] J. J. Monaghan. SPH without a tensile instability. *J. Comp. Phys.*, 159:290–311, 1999.

[63] P. W. Cleary & J. J. Monaghan. Boundary interactions and transition to turbulence for standard CFD problems using SPH. *Preprint.*

[64] J. P. Morris, P. J. Fox, and Y. Zhu. Modelling low Reynolds Number Incompressible Flows Using SPH. *J. Comp. Phys.*, 136:214–226, 1997.

[65] J. P. Morris and J. J. Monaghan. A Switch to Reduce SPH Viscosity. *J. Comp. Phys.*, 136:41–50, 1997.

[66] I. Nezu and H. Nakagawa. *Turbulence in Open-Channel Flows.* IAMR Monograph Series, 1993.

[67] F.T.N. Nieuwstadt, J.G.M. Janssen, and M.J.B. Pourquié. Direct and large-eddy simulations of turbulence in fluids. In *Amsterdam, the Netherlands*, 1993.

[68] P.A. Patsis and E. Athanassoula. SPH simulations of gas flow in barred galaxies. Effect of hydrodynamical and numerical parameters. *Astronomy and Astrophysics*, 358:45–56, 2000.

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

[70] O. Reynolds. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philos. Trans. R. Soc. London Ser.*, A 174:935–982, 1883.

[71] P. Rollet-Miet. *Simulation des grandes échelles sur maillage non-structuré pour géométries complexes.* PhD thesis, Laboratoire de Mécanique des Fluides et d'Acoustique de l'Ecole Centrale de Lyon, 1998.

[72] S. Shao and E. Y. M. Lo. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources*, 26:787–800, 2003.

[73] Z. Shao and S. A. Yost. A portable numerical method for simulating 2D unsteady free-surface flows. *Report of University of Kentucky, Department of Civil Engineering*, pages 1–16.

[74] J. Shi, T. G. Thomas, and J. J. R. Williams. Free-Surface Effects in Open Channel Flow at Moderate Froude and Reynold's Numbers. *Journal of Hydraulic Research*, 38(6):465–474, 2000.

[75] C. Da Silva and O. Métais. Vortex control of bifurcating jets: A numerical study. *Phys. Fluids*, 14, 2002.

[76] R. Speith and H. Riffert. The viscous gas ring as an astrophysical test problem for a viscous SPH-code. *J. Computational and Applied Mathematics*, 109:231–242, 1999.

[77] J. W. Swegle, D. L. Hicks, and S. W. Attaway. Smoothed Particle Hydrodynamics Stability Analysis. *J. Comp. Phys.*, 116:123–134, 1995.

[78] N. Takada, M. Misawa, A. Tomiyama, and S. Hosokawa. Simulation of Bubble Motion under Gravity by Lattice Boltzmann Method. *Journal of Nuclear Science and Technology*, 38:330–341, 2001.

[79] H. Tennekes and J. L. Lumley. *A first course in turbulence.* MIT Press, Cambridge, 1972.

[80] C. Tong and Z. Warhaft. Passive scalar dispersion and mixing in a turbulent jet. *J. Fluid Mech.*, 292:1–38, 1995.

[81] J.C. Uribe and D. Laurence. 10th Ercoftac/IAHR Workshop on Refined Turbulence Modelling. In *10th joint ERCOFTAC (SIG-15)/IAHR/QNET-CFD Workshop on Refined Turbulence Modelling*, 2000.

[82] M. Salinas Vázquez. Simulation des grandes échelles d'écoulements dans les canaux de refroidissement de moteur fusée, PhD thesis. *Institut National Polytechnique Grenoble*, 1999.

[83] M. Salinas Vázquez and O. Métais. Large-eddy simulation of the turbulent flow through a heated square duct. *J. Fluid Mech.*, 453:201–238, 2002.

[84] D. Violeau. Notions fondamentales de mécanique lagrangienne. *Rapport EDF-LNHE / HP-72/2000/072/A*, 2000.

[85] D. Violeau. Logiciel Spartacus-2D de modélisation lagrangienne des écoulements de fluides. Notice d'exploitation de la version 1.0. *Rapport EDF-LNHE / HP-75/2001/028/A*, 2001.

[86] D. Violeau. Instability criterion proof. Personnal e-mail, Electricité de France, 2002.

[87] D. Violeau. Application of Smoothed Particle Hydrodynamics to the design of sea defence. *IARH Newsletter*, 2003.

[88] D. Violeau. Les fondements de la turbulence. Cours LNHE, EDF, 2003.

[89] D. Violeau. One and two-equations turbulent closures for Smoothed Particle Hydrodynamics. In *6th International Conference on Hydroinformatics*, June 2004.

[90] D. Violeau, C. Chauvelier-Alario, and A. Foucart. Etude de faisabilité d'un prototype de logiciel fondé sur la méthode Smoothed Particle Hydrodynamics (SPH) pour la mécanique des fluides. *Rapport EDF-LNHE / HP-42/1999/065/A*, 1999.

[91] D. Violeau and R. Issa. La méthode numérique SPH appliquée a l'hydraulique. *Revue Européenne des Eléments Finis*, 12:171–190, 2003.

[92] D. Violeau and R. Issa. Modelling turbulent free surface flows with Smoothed Particle Hydrodynamics. *Ercoftac bulletin*, 2004.

[93] D. Violeau and S. Piccon. Prise en compte des effets turbulents dans la méthode numérique SPH. introduction d'un modèle de viscosité turbulente et d'un modèle stochastique. *Rapport EDF-LNHE / HP-75/02/005/A*, 2002.

[94] D. Violeau, S. Piccon, and J. P. Chabard. Two attempts of turbulence modelling in Smoothed Particle Hydrodynamics. *In advances in Fluid Modelling and Turbulence Measurements, World Scientific*, 2002.

[95] P. L. Viollet, J. P. Chabard, P. Esposito, and D. Laurence. *Mécanique des fluides appliquée. Ecoulements incompressibles dans les circuits, canaux, rivières, autour de structures et dans l'environnement.* Presses de l'Ecole Nationale des Ponts et Chaussées, 1998.

[96] W. C. Welton. Two-dimensional PDF/SPH simulations of compressible turbulent flows. *J. Comp. Phys.*, 139:410–443, 1998.

[97] A. P. Whitworth, A. S. Bhattal, J. A. Turner, and S. J. Watkins. Estimating density in smoothed particle hydrodynamics. *Astron. Astrophys.*, 301:929–932, 1995.