Development of Efficient Meshless Methods for Free Surface Flows

Valentina Lucente

A thesis submitted in partial fulfilment of the requirements
of the
Manchester Metropolitan University for the degree of
Master of Philosophy

Manchester
Metropolitan
University

Department of Computing, Mathematics
and Digital Technology

2014

Abstract

The aim of this thesis is to develop an SPH method free of post formulation corrections with application to the classical Dam-break problem. The analysis has been focused on the weakly compressible smoothed particle hydrodynamics (SPH) formulation of the fluid flow.

The discretization and implementation of classical Euler equations has been revised in order to avoid artificial and unphysical corrections commonly used in literature. Many other tools are also used to improve accuracy and performance features, such as a new Kernel Theory to achieve the lowest error and a higher order adaptive time integrator. The results obtained are encouraging either from the physical reliability or the performance time. The obtained simulations are fully comparable with the other already existing in literature, as well as the running time being very promising as it is of order of few hours in a single CPU machine.

Acknowledgements

I wish to express my appreciation and sincere gratitude to my supervisors **Professor Derek M. Causon** and **Professor Clive G. Mingham** for their supervision, guidance and support throughout my research studies and for having introduced me to this very interesting subject.

My special acknowledgements to **Professor Clive G. Mingham** for his assistance and scientific hints during meetings and our stimulating discussions. His precious advice helped me to enhance my skills and knowledge. I am indebted to him for his continuous encouragement and support either from scientific or personal point of view.

I also wish to thank **Dr. Jon Shiach** for his revision of this thesis.

I finally acknowledge the Examiners **Dr. Steven Lind** and **Dr. Benedict D. Rogers** for their thoughtful revision and remarkable comments.

I am grateful to my boyfriend **Alessandro** for his support, encouragement and love which were sources of energy and motivation throughout this period.

Contents

List of Acronyms

CFD	Computational Fluid Dynamics
CSM	Computational Solid Mechanics
CSPM	Corrective Smoothed Particle Method
FDM	Finite Differences Method
FEM	Finite Elements Method
FVM	Finite Volume Method
I-SPH	Incompressible Smoothed Particle Hydrodynamics
MLS	Moving Least Squares
MPS	Moving Particle Semi-implicit
ODE(s)	Ordinary Differential Equation(s)
PDE(s)	Partial Differential Equation(s)
RK	Runge-Kutta, or RKF for Runge-Kutta-Fehlberg
RKF45	Runge-Kutta-Fehlberg 4th-5th orders
SPH	Smoothed Particle Hydrodynamics
WCSPH	Weakly Compressible SPH

List of Figures

List of Tables

Outline of the thesis

The aim of this thesis is the development of an SPH method free of post formulation corrections, suitable for the treatment of problems concerning violent wave impacts against fixed structures.

The discretization of the fluid equations using SPH (with *kernel approximation*) is consistent as h (smoothing length) approaches to zero, see e.g. Vila (1999). On the other hand, in order to guarantee a good approximation of the equations, the number of fluid particles in a neighbourhood of radius h of a given particle, should be as high as possible. Unfortunately, in the applications, such a condition is not always satisfied: first of all, in order to compute the interactions among particles a high CPU time is required, secondly there can exist regions in which the particle concentration is not sufficient to guarantee a prefixed order of accuracy (a very frequent situation in violent impact problems).

To overcome bad behaviours arising from low accuracy approximations of the flow equations, many corrections have been introduced in literature, such as artificial viscosity, XSPH, tensile correction, etc. Unfortunately, these tools have a non-obvious physical meaning, and often require the use of constants whose values are determined in an empirical and non-rigorous way. For this reasons, throughout the work, the possibility to avoid the use of corrections without affecting the simulation quality was investigated. Rather, attention was focused on the theory of kernel approximation, examining the possibility to choose more appropriate functions in order to reduce the error as much as possible.

Another difficult task is the correct implementation of the boundary action. Initially, Monaghan (1992), as a model of the above mentioned forces, used the well-known Lennard-Jones potential (more properly related to molecular problems). As it will be shown, such tool may lead to instability phenomena due to the singularity of the potential in presence of collisions. As a consequence, a very small time-step for the ODEs discretization scheme is required, and therefore a significant CPU time.

Another possible approach consists of using ghost particles. In most cases, especially in presence of high pressures and velocities problems (as in the dam-break case), such particles do not exert sufficiently high forces in order to prevent fluid particles passing through the boundary. Such trouble is generally overcome by using several layers of ghost particles (usually two or three) to improve the repulsive effect. Nevertheless, it is not evident how to give a general criteria for a rigorous (and not empirical) dependence between the number of layers of ghost particles and the problem at hand.

The classical dam-break problem is a simplified example of a violent wave impact problem, however its implementation exhibits a wide number of problems from the hydrodynamical and numerical point of view. The aim is to point out, through a large set of simulations, the above mentioned problems and how to overcome them by using various tools and improvements that were introduced.

Either the classical dam-break problem or its obstacle version have been implemented by a brand-new implementation, obtaining simulations that are fully comparable with the others in literature (e.g. Gomez-Gesteira *et al.* (2010) and Liu *et al.* (2003)). As for the above discussed CPU time, a direct approach (i.e. a particle-particle interaction test) would require a number of operations of order N^2 (where N is the number of particles). To prevent such huge cost, the interaction list (IL) tool is generally introduced. The latter is a structure linking the particles interacting with a given fluid particle. In principle it would be necessary to update the IL at every time-step, but that would be computationally expensive. The key point is that it is possible to give a criterion to avoid the IL updating process at every time-step. In this way the implementation efficiency can be improved, obtaining in this way a performance time reduction greater than 50 percent.

Main results

The only way to perform a critical evaluation of the SPH approach, was to produce an *ex-novo* code based on a revised discretization of the fluid equations. Together with the accuracy, the relevant problem of the performance time has been taken into account. The difficult task in performing the simulation on a single-CPU laptop, has required the use of several tools for an efficient management of the bounded performance capabilities.
Among the features of the obtained implementation the following *original* achievements can be mentioned:

- Complete absence of unphysical corrections, such as XSPH, artificial viscosity, tensile correction etc., and empirically determined parameters,
- Use of high order integrator as Runge-Kutta-Fehlberg (4th-5th orders, RKF45), with a fully automated time-step adjustment,
- Use of a second order smoothing kernel leading to a minimal error (optimal kernel), see Sec 3.1.2,
- CSPM formulation of the Euler equations.

Chapter 1

Introduction and Literature Review

Smoothed Particle Hydrodynamics (SPH) is a recent development in the field of Computational Fluid Dynamics (CFD). It is a Lagrangian method that does not use a mesh, where the fluid is discretized into a set of particles, whose motion are explicitly computed and tracked via standard tools of Ordinary Differential Equations (ODEs). The main step leading to the particle approximation is the use of a smooth function, called smoothing-kernel, to define the collision rules (i.e. the interacting forces) among the particles.

As a meshfree technique, it facilitates the simulation of certain problems where Eulerian methods can be difficult to apply, such as violent wave impacting structures during storms, on coastal and fixed marine structures or breakwaters, harbour walls, cliffs, and other sea defences. Recent works, e.g. Hughes *et al.* (2008), have shown that a fully incompressible SPH formulation (I-SPH) can provide accurate simulations of wave impacts and their flip-through phenomena against the wall of some coastal structures.

The estimation of loads arising from the action of these violent wave impacts is a key point of these simulations, since it is essential to prevent inland flooding, damage to structures and to guarantee their survivability in such hostile conditions.

Few numerical methods have been widely used to treat this kind of problem, such as the classical finite difference, finite volume, or finite element methods, despite being considered nowadays as fundamental tools in CFD. However these methods are essentially mesh-based, and often exhibit some difficulties in the treatment of free-surface, moving or deformable boundary problems. The necessary process of generating and refreshing a mesh may require large CPU times. Moreover, in certain problems, the mesh capturing the free surface may break-up and recombine as the wave breaks, see Hughes *et al.* (2008), and this can be a very difficult task.

Existing methods such as SPH and MPS (Moving Particle Semi-implicit) use I-SPH or weakly compressible (WCSPH)single-phase formulation with a stiffened equation of state for the water. Both methods have been shown to be well suited to treat wave impacts with some common generic features. Nevertheless, the incompressible formulation does not work properly with so-called aerated impacts, that is, when the wave breaks against the wall. In any case the single-phase formulation implies the vanishing of enclosed air pockets produced during wave generation.

WCSPH uses explicit schemes that do not require solving the pressure Poisson equation. However, without any correction, the density of the particles rapidly exhibits some noisy and chaotic behaviour, implying, in this way, a loss of hydrodynamical information.

1.1 Numerical methods in hydrodynamics

As supercomputers and parallelization techniques have been developed in recent years, they have replaced expensive (and often very dangerous) laboratory experiments. As for the hydrodynamical problems, numerical simulations are appropriate as the latter are able to describe large-scale phenomena (such as tsunamis, inland floods, etc.) that cannot be studied via laboratory experiments. As a problem involving continuous media, a hydrodynamical model, is described by a system of Partial Differential Equations (PDEs). Therefore every simulation provides a finite set of ODEs, which characterizes the discretization technique (or *method*) at hand. Depending on the geometry of the problem and the evolution of the fluid mass, it is possible to solve the domain decomposition problem through a **Mesh-based** method or a **Meshless** method. Each of them could be suitable for a given problem but highly disadvantageous for another one. This choice strongly depends on the geometry of the problem and its evolution.

1.1.1 Mesh-based methods

There are two fundamental points of view to describe the fluid evolution: *Eulerian* and *Lagrangian*. See Sec. 1.3 for their mathematical description.
The first one uses mainly the Finite Differences Method (FDM) or Finite Volume Method (FVM) to discretize the system of PDE, while the second one, that uses a material description of the flow, is based on the Finite Elements Method (FEM). Both have been widely used in literature, see e.g. Anderson (1995), Hirsch (1988), Wilkins (1999) for the FDM, and Taylor *et al.* (2000), Liu *et al.* (2003) for the FEM.

Lagrangian mesh

As a main feature of the Lagrangian technique the mesh is attached to the material throughout the entire simulation, therefore the mesh cells move according to the material movement and their shape and volume can change during the computation process. This point of view has several advantages:

1. An irregular mesh can be used to treat conveniently complicated geometries,

2. The grid points are collocated only within the problem domain, implying CPU time saving,

3. The approach facilitates the treatment of free surface and moving boundary problems,

4. Easy tracking of all the field variables at a single material point (as the grid moves together with the material).

This method is very successful in problems where there is not a large material deformation. It is very popular in solving computational solid mechanics (CSM) problems.

However there could be problems when the mesh is very distorted, affecting the computational accuracy. This aspect has been pointed out in simulation of events of extremely large deformation, as in Hans (1999) and Benson (1992).

Eulerian mesh

Differently from the Lagrangian approach, the Eulerian mesh is fixed in space. The material moves across the mesh cells while their shape and volume remain unchanged for all t. This method could be suitable for large deformation problems as the mesh evolution is not required. The above described process has to be performed on the entire space, even though the fluid occupies a small portion of it, and this could be computationally expensive. In addition, the fluid profile is not exactly known (but only if a given cell is occupied or not), unless one resorts to a very fine mesh (with obvious computational costs).

Combination of Lagrangian and Eulerian mesh and limitation of grid-based methods

The above described techniques have been combined in order to overcome the difficulties of each one, obtaining in this way more stable solutions. During the last years specialists have tried to develop many commercial hydrocodes using this approach. See e.g. Hallquist (1988) and Hallquist (1998).
Nevertheless, in many situations, mesh-based numerical methods exhibit some difficulties. First of all, the mesh generation process could be very expensive, especially in large deformations, irregular or complex geometries and free-surface flows. Of course, the use of this class of methods is challenging for certain phenomena such as explosions and high velocity impact. In both cases, there exist large deformations, moving material interfaces, free surfaces and deformable boundaries that are difficult to treat via a grid-based approach.

1.1.2 Meshless methods

Meshless methods give a more appropriate approach to the above described class of phenomena, in which large errors or expensive simulation processes arise with grid-based techniques. The basic idea is to construct a moving mesh for a more stable and robust version of the standard grid-based methods. This point of view allows the simulation of a wide class of phenomena such as those arising from maritime and naval architecture, geotechnical, solids and fracture mechanics and biomechanics fields.

In this framework, Smoothed Particle Hydrodynamics (SPH) offers a flexible and powerful tool in order to implement the above described problems. After the pioneering works Lucy (1977) and Gingold *et al.* (1977), in Liszka *et al.* (1980) a generalized FDM that can deal with arbitrary irregular grids is proposed. Afterwards, many other methods have been improved and developed in order to solve some specific computational matters, such as consistency of SPH approximation close to the boundary (Reproducing Kernel Particle Method) or the singularity issue in the polynomial Point Interpolation Methods. Nowadays some substantial improvements have been achieved either from the theoretical or the simulation performance point of view. In Lind *et al.* (2012b) the multi-phase incompressible setting is developed. In particular, the gaseous phase is modelled with a compressible SPH formulation. The paper is a further improvement of Lind *et al.* (2012a), where the obtained pressure field with the I-SPH formulation is noise free and fully comparable with the semi-analytical solution.

1.2 Applications of meshless methods

As above mentioned, the class of problems in which the application of mesh-based methods exhibits technical issues is extremely wide. It should be stressed that in areas such as astrophysics, solid state physics, biophysics, biochemistry and biomedical research, one may encounter situations where the object under consideration is not a continuum, but a set of particles. It is clear that a discretization is not required in such cases and a particle-based method is the most natural choice for their numerical simulation. Forma-

tion phenomena of star systems, nano-scale movements of atoms in a non-equilibrium state, folding and unfolding of DNA are relevant example of such a class of problems.
On the other hand, there is a wide range of problems arising from continuum mechanics for which the discretization in terms of nodes turns out to be much more convenient either from the accuracy or computational cost aspects point of view. All the problems involving features such as large deformations, moving boundaries, complex geometries or detonation and collisions are well suited for a meshless based treatment.
Meshfree Galerkin methods (see e.g. Belytschko *et al.* (1996)) have clear advantages over finite element computations due to its ability to handle large deformation problems: crack growth/propagation problems Dolbow *et al.* (2000), large deformation of rubber materials Chen *et al.* (1997), metal forming and extrusion Chen *et al.* (1998) can be mentioned among the successful applications of this approach. The paper Adams *et al.* (2009) describes the advantages of the Moving Least Squares (MLS) method, able to provide high-order accuracy for elasticity computations in the context of computer graphics.

At the origin of the SPH theory, which is among the most consolidated and developed among the meshless methods, the contributions of Lucy (1977) and Gingold *et al.* (1977), involved astrophysical problems such as the formation and evolution of proto-stars or galaxies. Essentially, they observed that the collective movement of those particles is similar to the movement of a liquid, or gas flow, and it may be modelled by the governing equations of classical Newtonian hydrodynamics. The cosmological applications of this approach concern a wide range of problems: single and multiple detonations in white dwarfs, Garcia-Senz *et al.* (1999), cloud fragmentation and collisions, Durisen *et al.* (1986) and even the evolution of the Universe problem, Monaghan (1990).
As it is well known, the SPH method has been successful in several applications in CFD that would have been difficult or even unattainable with mesh-based methods. In Panizzo *et al.* (2004) underwater landslide generated waves are studied by using a parallel algorithm in order to optimize the performances of the model.
Another example of a relevant application in such area is described in Panizzo *et al.* (2006) and Rogers and Dalrymple (2008), where the SPH model allowed the study of a sliding

mass impacting a body of water.

The dam-break problem, discussed in the next chapters, can be interpreted as a toy model of a violent impact problem. Its implementation could be expensive and inaccurate using a mesh-based method (due to the small area occupied by the water, with respect to the whole domain), while it is naturally and profitably treated with the use of the SPH approach.

1.3 Eulerian and Lagrangian descriptions of motion

In this section the differences between the Lagrangian and Eulerian description of motion are addressed. This is a key step in order to introduce the SPH discretization technique.

Given a continuum system, a certain number of quantities of mechanical and/or thermodynamical nature, can be associated to it. These quantities, such as velocity or thermal energy in the case of a fluid, can be scalar or vectorial; in any case these are defined *locally* i.e. they depend on the considered point of the system.

There exist essentially two descriptions that can be used to represent a given quantity: *Eulerian* and *Lagrangian*.

In the first case the value of a chosen (e.g. scalar) quantity Q is a function of the time t and of the spatial coordinates $\underline{x} \in \mathbb{R}^3$ in a fixed frame of reference with respect to the moving fluid. In the second one, instead, the quantity Q (still a function of the time) depends on the so-called *material coordinates* $\underline{X} \in \mathbb{R}^3$. This is a comoving frame of reference with the fluid particles. More precisely, denoting with $\underline{\Phi}_t$ the fluid flow, the relation between the two coordinate systems is

$$\underline{X} = \underline{\Phi}_t^{-1}(\underline{x}). \tag{1.1}$$

In other words, in the Lagrangian description the observer follows a given particle in order to evaluate Q. In the Eulerian case Q is evaluated at a fixed point of the space, no matter which particle occupies that particular point during the evolution.

Let $Q_E(t, \underline{x})$ and $Q_L(t, \underline{X})$ the Eulerian and Lagrangian representation of the same quantity, respectively. Hence, by using (1.1) the relation between the two descriptions is

$$Q_L(t,\underline{X}) = Q_E(t,\underline{\Phi}_t(\underline{X})) \equiv Q_E(t,\underline{x}(t,\underline{X})). \tag{1.2}$$

The *material derivative* of a quantity Q describes its rate of variation as a function of time. It is immediate that in the Lagrangian description the material derivative coincides with the partial derivative with respect to time

$$\frac{DQ}{Dt}(t,\underline{X}) = \frac{\partial Q_L}{\partial t}(t,\underline{X}). \tag{1.3}$$

By (1.2), the computation of the same derivative in the Eulerian case, leads to a further term, due to the dependence of $\underline{x}$ on t. I.e.

$$\frac{DQ}{Dt}(t,\underline{X}) = \frac{\partial Q_E}{\partial t}(t,\underline{x}(t,\underline{X})) + \sum_{i=1}^{3} \frac{\partial Q_E}{\partial x_i}(t,\underline{x}(t,\underline{X}))V_i(t,\underline{X}), \tag{1.4}$$

with $V_i := \partial x_i/\partial t$. By using (1.1) we find

$$\frac{DQ}{Dt}(t,\underline{X})|_{\underline{X}=\Phi_t^{-1}(\underline{x})} = \frac{\partial Q_E}{\partial t}(t,\underline{x}) + \sum_{i=1}^{3} \frac{\partial Q_E}{\partial x_i}(t,\underline{x})v_i(t,\underline{x}) \tag{1.5}$$

where $v_i(t,\underline{x}) := V_i(t,\underline{X})|_{\underline{X}=\Phi_t^{-1}(\underline{x})}$ is the velocity field in the Eulerian description.

1.4 Why SPH?

SPH was initially developed to deal with non-asymmetric phenomena in astrophysics in Lucy (1977) and Gingold *et al.* (1977). Fields such as free-surface flows, where Eulerian methods can be difficult to apply, represent a very high potential of applications (waves, impact on dams, offshore etc.) as the meshfree technique facilitates the simulation of highly distorted fluids and bodies. In such phenomena the geometry can change very drastically losing the regularity properties of a structured scheme which enables a domain decomposition software. Furthermore, with the ever increasing size and cost reduction of computer clusters, parallel simulations allow large scale simulations that were previously limited to mainframes. The classical dam-break problem, that is going to be studied throughout this work, exhibits some various features for which the SPH approach is appropriate. For instance, the volume occupied by the fluid mass is just a small part of the problem domain. In addition the wave reaches high velocities within a small time interval. Finally, the wave impact leads to an irregular and complicated wave front that cannot be easily tracked with the standard tools of mesh-based methods.

After an overview of the numerical methods for PDEs and a description of the meshless methods advantages for a wide number of applications, in the next chapter, the basics of the SPH approach will be introduced.

In particular, the SPH approximation theory is recalled, together with the discretization of the fluid equations based on this approach. The final part of the chapter is devoted to the description of an adaptive Runge-Kutta scheme. The latter will be used for the treatment of the system of ODEs arising from the fluid equations discretization.

Chapter 2

Background of SPH method

This chapter is devoted to the background material. Firstly, some basic notions of the SPH approximation theory have been recalled, with some relevant remarks on its accuracy and on the error introduced in the approximation. This is a key step for a critical evaluation of the numerical consequences of the SPH approach.

After recalling the classical Euler equations of the inviscid fluids, the standard derivation of the SPH equations for the fluid flow will be described.

2.1 The SPH approach

2.1.1 Main ideas

In order to fix the simplest framework, the basic properties of the objects at hand are briefly recalled, restricting the treatment to the one dimensional case. The extension to higher dimension is straightforward.

The key idea in the SPH approach is the following. Let $[a, b]$ be a compact subset of $\mathbb{R}$, for all $k \geq 1$, denoting with $C^k([a, b], \mathbb{R})$ the set of $k-$times differentiable functions on $[a, b]$ with continuous derivative(s). Now, for all $f(x) \in C^k([a, b], \mathbb{R})$ the **Dirac delta** is implicitly defined by the following equality

$$\int_a^b \delta(x - x_0) f(x) := f(x_0). \tag{2.1}$$

The SPH approximation consists in replacing the above defined "function" (more properly, distribution, see for instance Vladimirov (1981)) with a smooth function[1]. So is given the following

Definition 2.1. *A* ***smoothing kernel*** *is defined as a function* $W(x,h) \in C^\infty([a,b],\mathbb{R})$ *satisfying, for all* $x_0 \in [a,b]$ *the following conditions*

1. $\displaystyle\int_a^b W(x - x_0, h)dx = 1,$ unity condition
2. $\displaystyle\lim_{h\to 0}\int_a^b W(x - x_0, h)dx = \int_a^b \delta(x - x_0)dx,$
3. $W(x - x_0, h) = 0$ *if* $|x - x_0| > kh$ compact support.

The real positive parameter h *is usually known as* smoothing length *and* k *is a scale factor.*

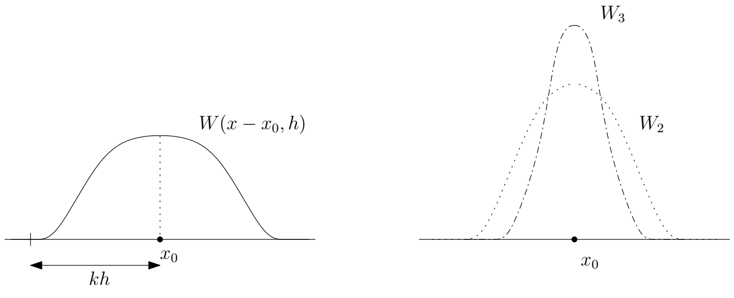


Figure 2.1: A typical smoothing kernel shape (left). On the right a pictorial idea of the Dirac delta definition as the weak limit of the sequence $W_n := W(x - x_0, h/n)$ for $n \to \infty$

In this setting, the first well known approximation formula of the SPH theory takes the form

$$f(x_0) = \int_a^b f(x)\delta(x - x_0)dx \sim \int_a^b f(x)W(x - x_0, h)dx. \tag{2.2}$$

[1] vice-versa, the Dirac delta can be defined as the (weak) limit of a sequence of smooth functions

The second approximation step in SPH can be described as follows. Consider a set of (randomly distributed) points $x_i \in (a, b)$, $i = 1, ..., N$ modelling the particle distribution at the generic time t. Then it is possible to construct a "lower approximating covering" of the interval $[a, b]$ by taking into account the union of certain sets I_i centred at x_i respectively, defined in a way these are as large as possible but non-intersecting. More precisely, let

$$h^* := \min_{i=1,...,N-1} \frac{x_{i+1} - x_i}{2}, \qquad h := \min\{x_1 - a, b - x_N, h^*\}, \tag{2.3}$$

by defining $I_i := [x_i - h, x_i + h]$ one has

$$\bigcup_{i=1}^{N} I_i \subseteq [a, b], \tag{2.4}$$

and the equality holds if, and only if, x_i are equally spaced on $[a, b]$ and the endpoints of I_1 and I_N match exactly with a and b respectively, as usually done in the classical Riemann integration theory (see Figure 2.2).

In this setting, the following approximation formula holds

$$\int_a^b f(x)dx \sim 2h \sum_{i=1}^{n} f(x_i). \tag{2.5}$$

It is clear that in the above described situation of equally spaced points, this formula is exactly the simplest quadrature formula used in numerical integration, often known as the *rectangle rule*.

The main drawback in choosing randomly distributed points clearly arises in dimension greater than one: if a domain decomposition of Ω realized by I_i is a union of disks or spheres that, even if defined with non-constant radius, cannot cover the entire domain unless the limit $N \to \infty$ is taken. The corresponding one-dimensional case is depicted in Figure 2.2.

As a matter of fact, in this particle-based approach, the only way to obtain a good approximation is to take N very large; in this case the distribution (if uniform) approximates an equal spacing and relations (2.4) and (2.5) tend to equality (exactly as done in Monte Carlo methods).

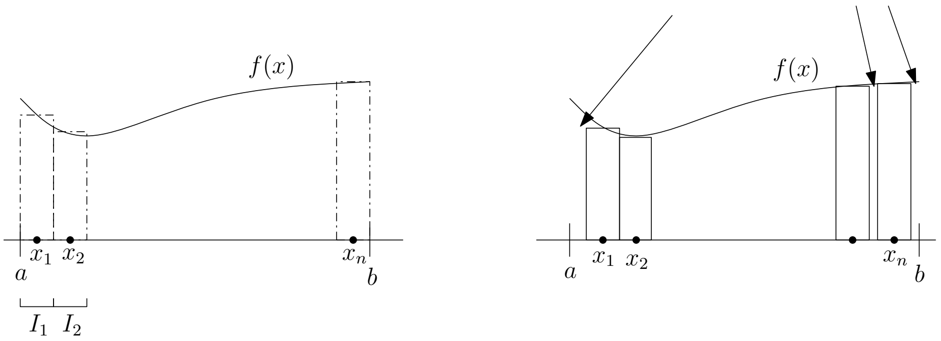


Figure 2.2: A comparison between the classical rectangle rule and the integration performed in the SPH approximation. The arrows show some missing areas due to the non-uniform spacing.

Let us visualize I_i in a three dimensional space: I_i is a volume centred at $\underline{\mathrm{x}}_i \in \mathbb{R}^n$ (a position vector), so it can be interpreted as the mass m_i of the particle divided by its density ρ_i. In this way, the second fundamental rule in SPH approximation for a general one, two or three dimensional domain Ω, is obtained

$$\int_\Omega f(\underline{x})d\underline{x} \sim \sum_{i=1}^{N} \frac{m_i}{\rho_i} f(\underline{x}_i). \tag{2.6}$$

The dependence of m_i and ρ_i upon i should be regarded as an adaptive method of covering Ω.

Remark 1. While the first formula (2.2) requires a small value of the smoothing length, the second one (2.6) requires a great number of particles within the subdomain I_i. As I_i, by construction, has to contain the compact support of $W(|\underline{x} - \underline{x}_i|, h)$, it is evident that a SPH simulation must provide a very large number of particles.

2.1.2 Accuracy and consistency of the SPH approximation

Now, if $f(x)$ is replaced with its Taylor expansion computed at x_0 in the right hand side of (2.2), one obtains

$$\begin{aligned}\int_a^b f(x)W(x-x_0,h)dx &= \int_a^b \left[f(x_0)+\sum_{n=1}^{+\infty}\frac{1}{n!}f^{(n)}(x_0)(x-x_0)^n\right]W(x-x_0,h)dx \\ &= f(x_0)+\sum_{n=1}^{+\infty}\frac{f^{(n)}(x_0)}{n!}\int_a^b W(x-x_0,h)(x-x_0)^n dx.\end{aligned} \tag{2.7}$$

By choosing $W(x-x_0)$ as an even function (i.e. symmetric with respect to x_0), with x_0 suitably chosen in a way that, for a given h,

$$[x_0-h, x_0+h] \subseteq [a,b] \tag{2.8}$$

the odd terms vanish. This implies

$$\int_a^b f(x)W(x-x_0,h)dx = f(x_0)+R(x_0,h) \tag{2.9}$$

$$R(x_0,h) = 2\sum_{n=1}^{+\infty}\frac{f^{(2n)}(x_0)}{(2n)!}\int_0^h W(r,h)r^{2n}dr. \tag{2.10}$$

Hence the error arising in the approximation (2.2) is of order h^2. The possibility to improve the approximation by reducing (or even neglecting) the remainder $R(x_0,h)$ up to an arbitrary high order in h by a careful choice of the kernel function will be discussed later.

Nevertheless, for a given h, all the $x_0 \in \{(a,a+h)\cup(b-h,b)\}$ (particles close to the boundary) do not satisfy (2.8). In these cases the integrals $\int_a^b W(x-x_0,h)(x-x_0)dx$ do not vanish and the error is of $O(h)$. This implies that the particles close to the boundary in the SPH approximation of the flow equations are affected by an error greater than the one of the internal particles. In addition, even if the hypothesis (2.8) is satisfied, the case of non-uniformly distributed particles in the support domain of the kernel, may lead to integrals that, computed as a weighted sum all over the particles, will be not zero. The error in computing integrals via formula (2.6) is not smaller than the error arising in the classical rectangle rule. It is well known that, if $f'(x)$ is bounded on $[a, b]$, this error is $O(h)$. So the convergence of the SPH approach could be very "slow" as $h \to 0$.

2.2 Derivation of SPH equations

In this section the SPH approach in order to discretize the inviscid fluids equations is recalled. This is a standard argument and it is used systematically in the SPH literature. Nevertheless, a brief discussion of this key procedure could be useful either to discuss the introduced approximation or to fix the notational setting once and for all.

Let $\Omega \subset \mathbb{R}^n$ be the domain surrounding the fluid. The classical Euler equations for inviscid fluids read as

$$\frac{D}{Dt}\underline{u} = -\frac{1}{\rho}\nabla P + \underline{F} \tag{2.11a}$$

$$\rho_t + \operatorname{div}(\rho\underline{u}) = 0 \tag{2.11b}$$

$$\underline{u}\cdot\underline{n} = 0 \qquad \text{on} \quad \partial\Omega \tag{2.11c}$$

where D/Dt is the material operator defined in (1.3), $\underline{u}(\underline{x}, t)$ is the velocity field at the point $\underline{x}$ and time t, $\underline{n}$ is the normal vector to the boundary $\partial\Omega$. P is a scalar function known as pressure and $\rho(\underline{x}, t)$ is the density. $\underline{F} = (F_1, F_2)$ is the resultant of the external forces.

Let N be the total number of fluid particles, and n_i be the number of particles interacting with the $i-$th particle, i.e. where distance from the $i-$th particle is not greater than kh.

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a smooth function. For a generic particle i, whose position is $\underline{x}_i$, the basic formula of the SPH approximation reads

$$\begin{aligned} f(\underline{x}_i) &\sim \int_\Omega f(\underline{x})W(|\underline{x}-\underline{x}_i|)d\underline{x} \\ &= \int_{\Omega_i} f(\underline{x})W(|\underline{x}-\underline{x}_i|)d\underline{x} \\ &\sim \sum_{j=1}^{n_i} f(\underline{x}_j)\frac{m_j}{\rho_j}W_j, \end{aligned} \tag{2.12}$$

where

$$\Omega_i := \{\underline{x} \in \Omega : |\underline{x} - \underline{x}_i| \leq kh\} \subset \Omega$$

is the kernel compact support.

The expression of the divergence of a function $\underline{g} : \mathbb{R}^2 \to \mathbb{R}^2$ can be written down by using the same formula[2]:

$$\begin{aligned} \operatorname{div} \underline{g}(\underline{x}_i) &\sim \int_{\Omega_i} \operatorname{div} \underline{g}(\underline{x}) W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &= -\int_{\Omega_i} \underline{g}(\underline{x}) \nabla W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &\sim -\sum_{j=1}^{n_i} \underline{g}(\underline{x}_j) \frac{m_j}{\rho_j} \nabla W_j \end{aligned} \tag{2.13}$$

Let us consider (2.11b), i.e. the continuity equation. The discretization can be obtained, for all $i = 1, ..., N$, by multiplying both sides of the latter equation by the smoothing kernel $W(|\underline{x} - \underline{x}_i|)$ and then integrating over Ω_i. In detail

$$\begin{aligned} \int_{\Omega_i} \partial_t \rho W(|\underline{x} - \underline{x}_i|) d\underline{x} &\sim \dot{\rho}_i \int_{\Omega_i} W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &\sim \dot{\rho}_i \left(\sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_j - \underline{x}_i|) \right) \end{aligned} \tag{2.14}$$

$$\begin{aligned} \int_{\Omega_i} \operatorname{div}(\rho \underline{u}) W(|\underline{x} - \underline{x}_i|) d\underline{x} &\sim -\rho_i \int_{\Omega_i} \underline{u} \cdot \nabla W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &\sim -\rho_i \sum_{j=1}^{n_i} \frac{m_j}{\rho_j} \underline{u}_j \cdot \nabla W(|\underline{x}_j - \underline{x}_i|) \end{aligned} \tag{2.15}$$

Note that the integral

$$\int_{\Omega_i} W(|\underline{x} - \underline{x}_i|) d\underline{x} \tag{2.16}$$

is (should be) equal to one according to the partition of unity.

Unfortunately, this condition is hardly ever satisfied in real simulations because of errors in numerical computation of the integral and non-uniform particles distribution. So this defect has been taken into account by bringing the SPH approximation of the integral (2.16) into the equations. This technique is usually known as *Corrective Smoothed Particle Method* (CSPM), see e.g. Liu *et al.* (2003).

[2] recall that $W(x - x_0)|_{\partial\Omega_i} = 0$

In this way, by (2.14) and (2.15), the continuity equation (2.11b) in SPH form reads:

$$\dot{\rho}_i = \left(\sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_i - \underline{x}_j|) \right)^{-1} \rho_i \sum_{j=1}^{n_i} \frac{m_j}{\rho_j} \underline{u}_j \cdot \nabla W(|\underline{x}_i - \underline{x}_j|). \tag{2.17}$$

The equation (2.11a) can be treated in the same way. The following identity can be used for its right hand side

$$\frac{1}{\rho} \nabla P = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho. \tag{2.18}$$

This trick is known as *symetrization* and was suggested by Monaghan (1988). In this way the right hand side splits into two terms,

$$\begin{aligned} \int_{\Omega_i} \nabla \left(\frac{P}{\rho} \right) W(|\underline{x} - \underline{x}_i|) d\underline{x} &= - \int_{\Omega_i} \frac{P}{\rho} \nabla W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &\sim - \sum_{j=1}^{n_i} \frac{P_j m_j}{\rho_j^2} \nabla W(|\underline{x}_i - \underline{x}_j|) \qquad (2.19) \\ \int_{\Omega_i} \frac{P}{\rho^2} \nabla \rho W(|\underline{x} - \underline{x}_i|) d\underline{x} &= - \frac{P_i}{\rho_i^2} \int_{\Omega_i} \rho \nabla W(|\underline{x} - \underline{x}_i|) d\underline{x} \\ &\sim - \frac{P_i}{\rho_i^2} \sum_{j=1}^{n_i} m_j \nabla W(|\underline{x}_i - \underline{x}_j|) \qquad (2.20) \end{aligned}$$

in addition

$$\int_{\Omega_i} \underline{F} W(|\underline{x} - \underline{x}_i|) d\underline{x} \sim \underline{F} \sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_i - \underline{x}_j|). \tag{2.21}$$

On the other hand

$$\int_{\Omega_i} \frac{D}{Dt} \underline{u} W(|\underline{x} - \underline{x}_i|) d\underline{x} \sim \dot{\underline{u}}_i \sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_i - \underline{x}_j|). \tag{2.22}$$

collecting the obtained terms, the momentum equation in the SPH form easily follows

$$\dot{\underline{u}}_i = \left(\sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_i - \underline{x}_j|) \right)^{-1} \sum_{j=1}^{n_i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W(|\underline{x}_j - \underline{x}_i|) + \underline{F}. \tag{2.23}$$

Denoting

$$\underline{z} \equiv (z^1, z^2, z^3, z^4, z^5) := (x_1, x_2, u_1, u_2, \rho),$$

equations (2.23) and (2.17) can be written in the following form

$$\dot{z}^i = \mathcal{F}^i(\underline{z}), \qquad i = 1, ..., 5. \tag{2.24}$$

where the vector field $\underline{\mathcal{F}}$ is explicitly given by

$$\left\{\begin{array}{rcl} \mathcal{F}^1 & = & z^3 \\ \mathcal{F}^2 & = & z^4 \\ \mathcal{F}^3 & = & \lambda_i \sum_{j=1}^{n_i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \frac{\partial}{\partial z_1} W(|\underline{x}_j - \underline{x}_i|) + F_1 \\ \mathcal{F}^4 & = & \lambda_i \sum_{j=1}^{n_i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \frac{\partial}{\partial z_2} W(|\underline{x}_j - \underline{x}_i|) + F_2 \\ \mathcal{F}^5 & = & \lambda_i \rho_i \sum_{j=1}^{n_i} \frac{m_j}{\rho_j} \underline{u}_j \cdot \nabla W(|\underline{x}_i - \underline{x}_j|) \end{array}\right. \tag{2.25}$$

with

$$\lambda_i := \left(\sum_{j=1}^{n_i} \frac{m_j}{\rho_j} W(|\underline{x}_i - \underline{x}_j|) \right)^{-1} . \tag{2.26}$$

Note that $\lambda_i \neq 0$ for all i as every particle interacts at least with itself. As for the pressure, as usual in SPH, the Tait equation of state has been used

$$P = P_0 \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right], \tag{2.27}$$

where $\gamma = 7$, ρ_0 is the density at $t = 0$ and P_0 is a linear function of height, suitably determined from the hydrostatic configuration.

2.3 Time integrator

2.3.1 Runge-Kutta VS lower order methods

The aim of this section is to give an high order time integrator for the already obtained system of ODEs. More precisely, the Adaptive Runge-Kutta-Fehlberg Method usually known as RKF45 is going to be described. As it is well known, Verlet and leap-frog schemes are widely used methods for SPH simulations. This is because of the compromise between computational cost (i.e. function evaluations) and accuracy. As an advantage, these schemes are *symplectic* and consequently they preserve the total energy of the system, see e.g. Sanz-Serna (1992). That is why, more in general, their use in Hamiltonian dynamics simulations looks to be so appropriate.

Unfortunately, in this case, the huge computational cost of a single evaluation of the vector field given by the (2.24) right hand side should be taken into account. The key fact is that, despite the (linearly) low number of function evaluations in case of a low order method, a (polynomially) higher number of time-steps is necessary in order to cover the same time interval, if compared with an higher order method. More precisely, in the case at hand, let $[a, b]$ be the time interval and $\varepsilon > 0$ the prefixed error.

The error of a Verlet scheme is $\mathcal{E}_v = C_v \sigma^2$ where C_v is a positive constant and σ denotes the time-step, while the number of the function evaluations at each stage is 2. Hence the total number of function evaluation N_v in order to reach the precision requested with ε, is obtained by setting $\sigma = (\varepsilon / C_v)^{\frac{1}{2}}$ then

$$N_v \geq 2 \left(\frac{C_v}{\varepsilon} \right)^{\frac{1}{2}} (b - a). \tag{2.28}$$

As for a non adaptive $4-$th order Runge-Kutta we have $\mathcal{E}_{rk} = C_{rk} \sigma^4$, then $\sigma = (\varepsilon / C_{rk})^{\frac{1}{4}}$ and finally

$$N_{rk} \geq 4 \left(\frac{C_{rk}}{\varepsilon} \right)^{\frac{1}{4}} (b - a), \tag{2.29}$$

where subscripts v and rk denote Verlet and Runge-Kutta, respectively.

By comparing these two formulae, it is clear the linear dependence on the function evaluation per stage versus the polynomial dependence on ε.

The RKF45 not only allows us to improve this aspect, but provides an automatic time-step choice by a joint action of a fourth order and a fifth order methods. This is a necessary approach when dealing with impact problems, in which the velocity field may lead to large variations in magnitude and a fixed step size could be unsuitable for certain parts of the simulation as redundant for another.

A general (explicit) $m-$stages Runge-Kutta (RK) method can be written in the form

$$\underline{x}_{i+1} = \underline{x}_i + \sigma \sum_{r=1}^{m} c_r \underline{k}_r, \tag{2.30}$$

where $\underline{k}_r = \underline{k}_r(t_i, \underline{x}_i, \sigma)$ are given by

$$\underline{k}_1 = \underline{f}(t, \underline{x}), \qquad \underline{k}_r = \underline{f}(t + \sigma a_r, \underline{x} + \sigma \sum_{s=1}^{r-1} b_{rs} k_s), \quad r = 2, ..., m, \tag{2.31}$$

and

$$\begin{cases} \dot{\underline{x}}(t) & = & \underline{f}(\underline{x}, t) \\ \underline{x}(t_0) & = & \underline{x}_0 \end{cases} \tag{2.32}$$

is the Cauchy problem at hand, while σ is meant to be interpreted as the *time-step*.

In order to construct a method is necessary to fix m and to determine the *Butcher tableau* of the Table 2.1.

0					
a_2	b_{21}				
a_3	b_{31}	b_{32}			
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
a_m	b_{m1}	b_{m2}	$\cdots$	b_{mm-1}	
	c_1	c_2	$\cdots$	c_{m-1}	c_m

Table 2.1: The Butcher tableau

The main argument allowing us to get an estimate of the truncation error every step, is to compare solutions computed by two (RK) methods of different order (e.g. 4 and 5).

Unfortunately, if one tries to do this by using two generic RK methods, as their respective tableau are completely different, the entire set of function evaluations of both schemes (at least $4+6=10$ in this case) turns out to be necessary.

The main feature of the RKF45 is to use a common set of function evaluations for both methods. More precisely, the single step of a $4-$th and a $5-$th order RK methods can be respectively performed by using the following formulae

$$\underline{x}_{i+1} = \underline{x}_i + \sigma \sum_{r=1}^{5} c_r k_r, \qquad \hat{\underline{x}}_{i+1} = \hat{\underline{x}}_i + \sigma \sum_{r=1}^{6} \hat{c}_r k_r \tag{2.33}$$

where

$$c_1 = \tfrac{25}{216}, \quad c_2 = 0, \quad c_3 = \tfrac{1408}{2565}, \quad c_4 = \tfrac{2197}{4104}, \quad c_5 = -\tfrac{1}{5},$$
$$\hat{c}_1 = \tfrac{16}{135}, \quad \hat{c}_2 = 0, \quad \hat{c}_3 = \tfrac{6656}{12825}, \quad \hat{c}_4 = \tfrac{28561}{56430}, \quad \hat{c}_5 = -\tfrac{9}{50}, \quad \hat{c}_6 = \tfrac{2}{55}$$

and the coefficients a_i and b_{ij} are given by the following common tableau

$1/4$	$1/4$				
$3/8$	$3/32$	$9/32$			
$12/13$	$1932/2197$	$-7200/2197$	$7296/2197$		
1	$439/216$	-8	$3680/513$	$-845/4104$	
$1/2$	$-8/27$	2	$-3544/2565$	$1859/4104$	$-11/40$

bringing down the function evaluations per step to 6.

2.3.2 Adaptive scheme

The aim is now to describe a criterion for a suitable choice of the time-step size $\sigma = \sigma_i$, in order to control the truncation error at each step[3] i.

[3] Obviously, the argument that is going to be illustrated for a $4-5-$th order methods, can be repeated whenever two numerical methods, which order is p and $p+1$, for all $p \geq 1$, are considered.

Let us consider a single step given by the two schemes (2.33) with the same initial data

$$\underline{x}_i = \hat{\underline{x}}_i = \underline{x}(t_i) \tag{2.34}$$

where i is chosen once and for all and $\underline{x}(t_i)$ denotes the exact solution at the time t_i. By definition, the truncation error ε_i of the first method is

$$\sigma_i \varepsilon_i = \underline{x}(t_{i+1}) - \underline{x}(t_i) - \sigma_i \sum_{r=1}^{5} c_r \underline{k}_r. \tag{2.35}$$

By construction

$$\underline{x}(t_{i+1}) - \underline{x}(t_i) - \sigma_i \sum_{r=1}^{5} c_r \underline{k}_r = \underline{x}(t_{i+1}) - \underline{x}_i - \sigma_i \sum_{r=1}^{5} c_r \underline{k}_r = \underline{x}(t_{i+1}) - \underline{x}_{i+1}. \tag{2.36}$$

In this way[4]

$$\begin{aligned} \varepsilon_i &= \frac{1}{\sigma_i}||\underline{x}(t_{i+1}) - \underline{x}_{i+1}||_\infty \\ &= \frac{1}{\sigma_i}||\underline{x}(t_{i+1}) - \hat{\underline{x}}_{i+1}||_\infty + \frac{1}{\sigma_i}||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty \\ &= \hat{\varepsilon}_i + \frac{1}{\sigma_i}||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty \end{aligned} \tag{2.37}$$

then ε_i is interpreted as a function $\varepsilon_i = \varepsilon_i(\sigma_i)$.

By assumption $\varepsilon_i = \alpha_i \sigma_i^4$ and $\hat{\varepsilon}_i = \beta_i \sigma_i^5$ for some $\alpha_i, \beta_i \in \mathbb{R}$. By substituting into the previous equation one has

$$\alpha_i \sigma_i^4 = \beta_i \sigma_i^5 + \frac{1}{\sigma_i}||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty \tag{2.38}$$

by multiplying both members by σ_i^{-4}, and for σ_i sufficiently small,

$$\alpha_i \sigma_i^4 = \varepsilon_i \sim \frac{1}{\sigma_i}||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty. \tag{2.39}$$

Now a corrected step $\sigma_i \leftarrow \gamma_i \sigma_i$ is searched, where γ_i is a scaling factor to be determined, in order to obtain

$$|\varepsilon_i(\gamma_i \sigma_i)| \leq \mathcal{E} \tag{2.40}$$

where $\mathcal{E}$ is the prefixed truncation error. On the other hand

$$\varepsilon_i(\gamma_i \sigma_i) = \alpha_i \cdot (\gamma_i \sigma_i)^4 = \gamma_i^4 \cdot (\alpha_i \sigma_i^4) \sim \frac{\gamma_i^4}{\sigma_i}||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty, \tag{2.41}$$

[4] If $\underline{v}$ is a (real-valued) $n-$dimensional vector, is denoted with $||\underline{v}||_\infty := \sup_{i=1,..,n} |v_i|$

where approximation (2.39) has been used in the last passage. Comparing with (2.40) the following estimate is finally obtained

$$\gamma_i \leq (\mathcal{E}\sigma_i)^{\frac{1}{4}} ||\hat{\underline{x}}_{i+1} - \underline{x}_{i+1}||_\infty^{-\frac{1}{4}}. \tag{2.42}$$

The latter has been used in the implementation as a time-step adjustment device.

The aim of this chapter was to introduce the standard tools and methodology of the SPH theory that is used in the implementation of the dam-break problem discussed in chapter 4.

The overview of the theory will be completed in the next chapter, addressing some topics related to the accuracy of the kernel approximation. In particular, the construction of higher order kernels described in Liu *et al.* (2003), is extended to all orders. This discussion is followed by the presentation of a new class of (optimal) kernels, which construction is based on the minimization of the approximation error.

A discussion on the ideas and the advantages of an interaction list storing criterion, originally due to Verlet, closes the rest of the next chapter. The performance features of this tool are compared with other results in literature.

Chapter 3

Theory

3.1 Construction of kernels

The kernel choice is a key step in the SPH approach. In this section, the possibility to improve the accuracy of the SPH approximation by using a kernel constructed in a suitable way, is discussed.

First of all, it will be shown that an arbitrary small interpolation error can be obtained only by requiring that the kernel is of non-constant sign in its domain. Unfortunately these kind of functions are unsuitable from a physical point of view.

In the light of this difficulty, a possible approach to obtain the kernel reaching the optimal accuracy is discussed. This kernel in the cubic case is explicitly computed.

3.1.1 Higher-order kernels

The aim is to construct a kernel approximating the Dirac delta up to an arbitrary high order. The approach goes along the lines of Liu *et al.* (2003) with some slightly different details. In addition, a general formula to construct a kernel of any order is given. As this discussion is of purely theoretical interest, the one-dimensional case is the only one that is going to be treated.

Let us consider a smooth function $f : [a, b] \to \mathbb{R}$ and let $h > 0$ and $x_0 \in (a, b)$ be such that condition (2.8) is satisfied.

Now recalling formula (2.10), express the remainder $R(x_0, h)$ as a sum of two terms

$$R(x_0, h) = R^{\leq N}(x_0, h) + O(h^{2N+2}), \tag{3.1}$$

where

$$R^{\leq N} := 2\sum_{n=1}^{N} \frac{f^{(2n)}(x_0)}{(2n)!} \int_0^h W(r, h) r^{2n} dr. \tag{3.2}$$

Now, the aim is to neglect the *cut-off*

$$R^{\leq N}(x_0, h) = 0 \tag{3.3}$$

obtaining, in this way, an accuracy of order $2N + 2$ with an arbitrary high $N \in \mathbb{N}$ that is supposed to be fixed.

The vanishing conditions arising from (3.3) together with the normalization condition, immediately lead to the following system

$$\begin{cases} \int_0^h W(r, h) dr & = & 1/2 \\ \int_0^h W(r, h) r^{2n} dr & = & 0 & \qquad \forall n = 1, ..., N \end{cases} \tag{3.4}$$

As the kernel is supposed to be an analytic function, it is constructed as a formal power series expansion in which the coefficients are determined as a solution of the already obtained system. As $W(r, h)$ is defined on a compact support, let us set $R := r/h$. The **polynomial kernel of degree** p can be naturally constructed as follows

$$W(R, h) := \begin{cases} (1 - R) \sum_{k=0}^{p-1} a_k R^k & \qquad R \in [0, 1] \\ 0 & \qquad R \geq 1 \end{cases} \tag{3.5}$$

In Liu *et al.* (2003) the vanishing condition of $W(R, h)$ at $R = 1$ is not intrinsically taken into account as above. This leads to an additional equation to the system (3.4).

First of all, note that (3.4) can be easily written as follows

$$\int_0^1 W(R, h) R^{2n} dR = \delta_n, \qquad n = 0, ..., N. \tag{3.6}$$

where $\delta_n = 1/2h$ if $n = 0$ and zero otherwise.

By taking into account of (3.5) one gets, for all $n = 0, ..., N$

$$\begin{aligned} \delta_n &= \int_0^1 \left(a_0 - a_{p-1}R^p + \sum_{k=1}^{p-1}(a_k - a_{k-1}) \right) R^{2n} dR \\ &= \frac{a_0}{2n+1} - \frac{a_{p-1}}{p+2n+1} + \sum_{k=1}^{p-1} \frac{a_k - a_{k-1}}{k+2n+1}. \end{aligned} \tag{3.7}$$

Now, collecting the pairs of terms a_k with the same subscript, after a re-summation, the following set of $N+1$ linear equations in the p unknowns $a_0, \ldots, a_{p-1}$ is obtained

$$\sum_{k=1}^{p} \frac{1}{(2n+k)(2n+k+1)} a_{k-1} = \delta_n, \qquad n = 0, ..., N. \tag{3.8}$$

This is clearly a square system if $N = p-1$, so a kernel of degree $p-1$ leads to an error of order $2p$, for all p.

The obtained system can be written in the usual form

$$Ax = b, \tag{3.9}$$

simply by defining

$$a_{ij} = \frac{1}{(2i+j-2)(2i+j-1)}, \qquad x := \begin{pmatrix} a_0 \\ \vdots \\ a_p \end{pmatrix} \qquad b := \begin{pmatrix} (2h)^{-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.10}$$

where a_{ij} are the entries of A.

As can be easily seen, the matrix A possesses an interesting entries structure (see also the next example).

The difficult problem of the computational aspects of the linear system (3.9) solution with an arbitrary high dimension is not here addressed. Due to the "bad" dependence of the a_{ij} as i and j increase, the matrix A turns out to be ill-conditioned, even worse than the classical Hilbert matrix[1].

[1] which elements are given by $h_{ij} := \frac{1}{i+j-1}$

Example 3.1.1 (Construction of a cubic kernel)**.** *What is needed is to solve system (3.9) in dimension three, where explicitly*

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{20} & \frac{1}{30} \\ \frac{1}{30} & \frac{1}{42} & \frac{1}{56} \end{pmatrix} \tag{3.11}$$

admitting as a solution

$$(a_0, a_1, a_2) = \left(\frac{15}{4h}, -\frac{57}{4h}, \frac{12}{h}\right). \tag{3.12}$$

The obtained kernel is then

$$W(R,h) = \frac{1}{4h}(1-R)(15-57R+48R^2). \tag{3.13}$$

It is easy to explicitly check that

$$2h\int_0^1 W(R,h)dR = 1, \qquad \int_0^1 W(R,h)R^k dR = 0, \quad k = 2, 4 \tag{3.14}$$

while

$$\int_0^1 W(R,h)R^6 dR = \frac{1}{210} \neq 0 \tag{3.15}$$

as expected.

It is clear that the kernel is not positive on the entire support. As pointed out in Liu *et al.* (2003), pag.85, this phenomenon leads to a non-physical situation such as negative pressures and energies. Unfortunately this obstruction is unavoidable at the first step too, as the integral

$$\int_0^h W(r,h)r^2 dr \tag{3.16}$$

cannot be neglected if $W(r,h)$ does not change sign.

3.1.2 Optimal kernels

Note that the integral appearing in formula (3.16) cannot be required to vanish. Nevertheless, it is possible to make it as small as possible via a careful kernel choice. This integral can be interpreted as a cost function and the coefficients of the polynomial kernel are determined as the solution of an optimization problem.

Another crucial feature of kernel functions is the behaviour of their second derivative on its support. As pointed out in Swegle *et al.* (1995) a second derivative changing of sign leads to certain instability region in the kernel support, implying some non-physical particle behaviours throughout the simulation. A way to overcome this trouble is the so-called Tensile correction (see, for instance Crespo (2008)). As usual, this requires parameters to be determined empirically like the average particle spacing.

In order to avoid the use of this correction, let us start with a kernel prototype whose second derivative does not change sign at all on the support.

Let $W = W(\rho)$ be the kernel function, where $\rho := |\underline{x} - \underline{x}_0|$. The compact support of $W(\rho)$ is a disk centred at $\underline{x}_0$ of radius l.

A suitable choice could be such that

$$W''(\rho) = a_0(l - a_1\rho), \qquad \rho \in [0, l], \tag{3.17}$$

with $a_0 \in \mathbb{R}^+$ to be determined. The previous function is positive for all $a_1 \in [0, 1]$.

As usual in kernel construction, in order to ensure a smooth joint with the exterior domain, the following vanishing conditions

$$W(l) = 0, \qquad W'(l) = 0, \tag{3.18}$$

are imposed. Firstly

$$W'(\rho) = a_0 \left(-\frac{1}{2}a_1\rho^2 + l\rho + a_2 \right), \tag{3.19}$$

for all $a_2 \in \mathbb{R}$. From (3.18b)

$$a_2 = l^2 \left(\frac{1}{2}a_1 - 1 \right). \tag{3.20}$$

Similarly, for all $a_3 \in \mathbb{R}$, follows

$$W(\rho) = a_0 \left(-\frac{1}{6}a_1\rho^3 + \frac{1}{2}l\rho^2 + a_2\rho + a_3 \right). \tag{3.21}$$

Now by taking into account of (3.20), condition (3.18a) gives

$$a_3 = l^3 \left(\frac{1}{2} - \frac{1}{3} a_1 \right). \tag{3.22}$$

In conclusion one gets

$$W(\rho) = -\frac{1}{6} a_0 [a_1 \rho^3 - 3l\rho^2 - 3l^2(a_1 - 2)\rho - l^3(3 - 2a_1)]. \tag{3.23}$$

The following statement is going to be used

Proposition 3.1.2 (Second order Taylor formula in $2D$)**.** *Let $W(\rho)$ be a kernel and $f(\underline{x}) \in C^2(\Omega, \mathbb{R})$, then the following formula holds*

$$\int_\Omega f(\underline{x}) W(\underline{x}) d\underline{x} = f(\underline{0}) + \frac{\pi}{2} \left(\frac{\partial^2 f}{\partial (x_1)^2}(\underline{0}) + \frac{\partial^2 f}{\partial (x_2)^2}(\underline{0}) \right) \int_0^l W(\rho) \rho^3 d\rho + o(k^2). \tag{3.24}$$

Proof. Given in Appendix □

It is clear that, in order to reach the best approximation, a minimization of the integral $I_3 := \int_0^l W(\rho)\rho^3 d\rho$ under the constraint

$$(2\pi)^{-1} = \int_0^l W(\rho) \rho d\rho =: I_1, \tag{3.25}$$

(unity condition) is required.

These integrals evaluated on kernel (3.23) give

$$I_1 = \frac{l^5}{6} a_0 \left(\frac{1}{4} - \frac{a_1}{5} \right), \qquad I_3 = \frac{l^7}{20} a_0 \left(\frac{1}{6} - \frac{a_1}{7} \right). \tag{3.26}$$

By substituting a_0 as determined by the unity condition (3.25)

$$a_0 = \left[\frac{\pi}{3l^5} \left(\frac{1}{4} - \frac{a_1}{5} \right) \right]^{-1}, \tag{3.27}$$

the optimization problem is reduced to

$$\min_{a_1 \in (0,1]} I_3 = \frac{l^2}{14\pi} \min_{a_1 \in (0,1]} \frac{7 - 6a_1}{5 - 4a_1}. \tag{3.28}$$

It is easy to check that the minimum is attained on the boundary at $a_1 = 1$. By (3.27), follows $a_0 = 60/\pi l^5$, implying that the best polynomial kernel of degree three is

$$W(\rho) = \frac{10}{\pi l^5}(l^3 - \rho^3 + 3l\rho^2 - 3l^2\rho). \tag{3.29}$$

Now replacing the smoothing length in its usual form $l = kh$ and performing the standard variable rescaling $R := \rho/h$, one obtains

$$W(R) = \frac{10}{\pi k^5 h^2}(k - R)^3, \qquad R \in [0, k] \tag{3.30}$$

and, obviously, $W(R) = 0$ if $R > k$.

In the simulations described in the following chapters, the previous kernel with $k = 2$ (as done, e.g. in Crespo (2008)) is used, i.e.

$$W(R) = \frac{5}{16\pi h^2}(2 - R)^3. \tag{3.31}$$

A comparison between this kernel and other classical kernels, such as cubic spline, quadratic and Gaussian, will be given in Section 3.2.

This argument can be generalized, in principle, to polynomial kernels with an arbitrary high degree or piecewise kernels. The substantial difference with the previous section is that an higher degree kernel leads to non-linear equations in the coefficients a_i. So the minimization problem easily solved in this very special case involves more complicated objects. A numerical solution of the implicit equations arising in this way is, in general, necessary.

An investigation of this possibility could be taken up in future work.

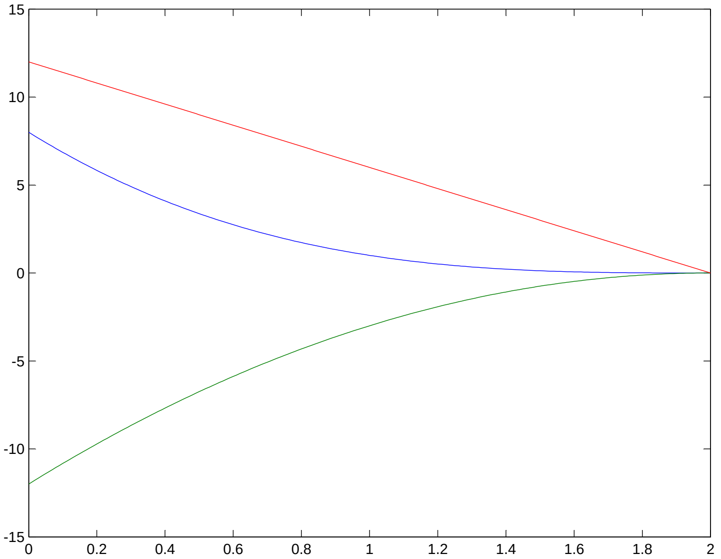


Figure 3.1: The optimal cubic kernel (divided by the factor $5/(16\pi h^2)$) depicted in blue, its first and second derivative in green and in red respectively. Note that $W''(R)$ is always non-negative on the compact support as required by construction.

3.2 Comparison between kernels

In this section a comparison between the kernel described in this work and the other classical kernels of the SPH literature is carried out. As already stressed in section 3.1.2 the interpolation error is of the form αh^β where the exponent $\beta \leq 2$ cannot in general be improved. The possibility to obtain the smallest constant α has been shown in section 3.1.2 as well. The aim is now to compare this constant with the other kernels by performing explicit computations on a generic function.

As a test function, let us consider the quadratic form $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x, y) = a_0 + b_1 x + b_2 y + c_{11} x^2 + 2c_{12} xy + c_{22} y^2, \tag{3.32}$$

with real coefficients. It should be stressed that the latter is the most general case to be considered, as the cubic terms would not contribute to the coefficient of the quadratic error.

In order to compute this coefficient, the following integral needs to be evaluated

$$I := \int \int_\Omega W_{\text{opt}}(x, y; h) f(x, y) dxdy, \tag{3.33}$$

where $W_{\text{opt}}(x, y; h)$ is the kernel defined by (3.31). In general, the error introduced by the kernel approximation is given by

$$\mathfrak{E} := I - f(0, 0). \tag{3.34}$$

By using polar coordinates for two dimensions, it reads as

$$\begin{aligned} I_{opt} &= \frac{5}{16\pi h^2} \int_0^{2h} \rho d\rho \int_0^{2\pi} d\theta f(\rho, \theta) \left(2 - \frac{\rho}{h}\right)^3 \\ &= \frac{5}{16\pi h^2} \int_0^{2h} \rho d\rho (2\pi a_0 + \pi c_{12} \rho^2 + \pi c_{22} \rho^2) \left(2 - \frac{\rho}{h}\right)^3 \\ &= a_0 + \frac{2}{7} h^2 (c_{12} + c_{22}). \end{aligned} \tag{3.35}$$

The same computation can be repeated below for the other kernels. These are recalled below for the reader's convenience (the same notational setting of Sec 3.1.2 is used).

1. Cubic spline

$$W_c(R,h) := \frac{10}{7\pi h^2}\begin{cases} 1-\frac{3}{2}R^2+\frac{3}{4}R^3 & \text{if} \quad 0 \le R < 1 \\ \frac{1}{4}(2-R)^3 & \text{if} \quad 1 \le R < 2 \\ 0 & \text{if} \quad R \ge 2 \end{cases} \tag{3.36}$$

2. Quadratic

$$W_q(R,h) = \frac{1}{\pi h^2}\left(\frac{3}{8}R^2 - \frac{3}{2}R + \frac{3}{2}\right), \qquad 0 \le R \le 2, \tag{3.37}$$

   and zero otherwise.

3. Gaussian

$$W_g = \frac{1}{\pi h^2}e^{-R^2}, \qquad 0 \le R \le 2, \tag{3.38}$$

   and zero otherwise.

By using these formulae, the same computation done for the Optimal kernel can be easily carried out as follows.

- Cubic spline:

$$\begin{aligned} I_c &= \frac{10}{7\pi h^2}\int_0^h \rho d\rho(2\pi a_0 + \pi c_{12}\rho^2 + \pi c_{22}\rho^2)\left(1 - \frac{3}{2}\frac{\rho^2}{h^2} + \frac{3}{4}\frac{\rho^3}{h^3}\right) + \\ &\quad \frac{5}{14\pi h^2}\int_h^{2h} \rho d\rho(2\pi a_0 + \pi c_{12}\rho^2 + \pi c_{22}\rho^2)\left(2 - \frac{\rho}{h}\right)^3 \\ &= a_0 + \frac{31}{98}h^2(c_{12}+c_{22}) \end{aligned} \tag{3.39}$$

- Quadratic:

$$\begin{aligned} I_q &= \frac{2}{\pi h^2}\int_0^{2h} \rho d\rho(2\pi a_0 + \pi c_{12}\rho^2 + \pi c_{22}\rho^2)\left(\frac{3}{4} - \frac{3}{4}\frac{\rho}{h} + \frac{3}{16}\frac{\rho^2}{h^2}\right) \\ &= a_0 + \frac{2}{5}h^2(c_{12}+c_{22}). \end{aligned} \tag{3.40}$$

- Gaussian:

$$\begin{aligned} I_g &= \frac{e^4}{(e^4-1)\pi h^2}\int_0^{2h} \rho d\rho(2\pi a_0 + \pi c_{12}\rho^2 + \pi c_{22}\rho^2)e^{-\frac{\rho^2}{h^2}} \\ &= a_0 + \frac{e^4-5}{2e^4-2}h^2(c_{12}+c_{22}). \end{aligned} \tag{3.41}$$

It is evident that, in each case, the errors $\mathfrak{E}$ are $O(h^2)$ as previously discussed. The corresponding coefficients of h^2 are computed numerically in the table below and graphically represented in Fig. 3.2.

Kernel	h^2 coefficient of $\mathfrak{E}$
Optimal	0.2857
Cubic spline	0.3163
Quadratic	0.4
Gaussian	0.4627

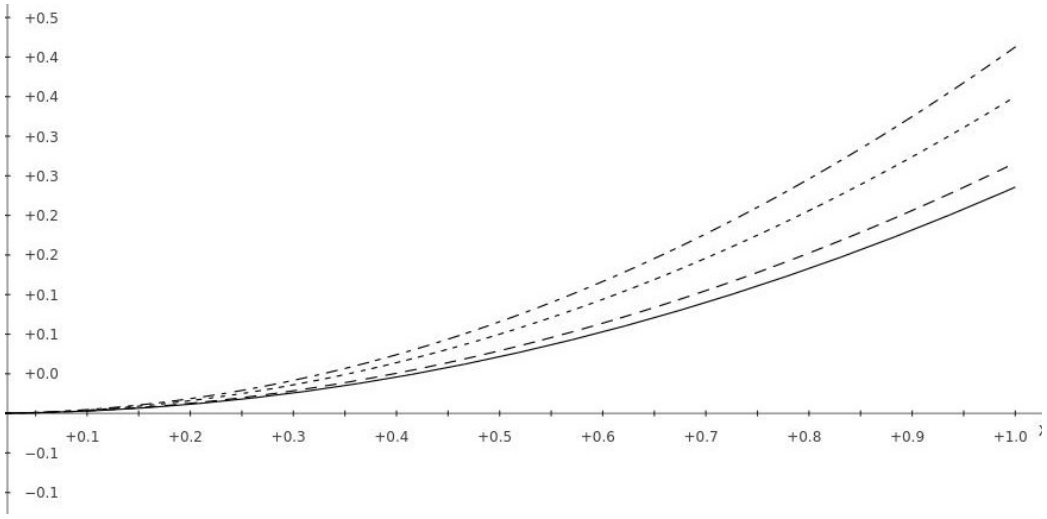


Figure 3.2: Comparison between the errors introduced by the above mentioned kernels: Optimal (continuous line), Cubic spline (dashed), Quadratic (dotted) and Gaussian (dash-dotted).

3.3 Interaction list (IL)

3.3.1 Set-up of the problem

The particle-based approach of the SPH is closely related to the computation of the interactions among particles. It is clear from the previously obtained discretized equation, that the motion of a particle is determined by means of neighbouring particles action, as weighted via the chosen kernel function.

The set of particles interacting with every single particle is stored in a structure called *interaction list* (IL). This can be considered as a matrix-like structure in which the number of rows is equal to the number of particles N, while the number of columns is not fixed. More precisely, it depends on the number of particles contained in a disk of radius equal to $l = hk$ and centered at the chosen particle.

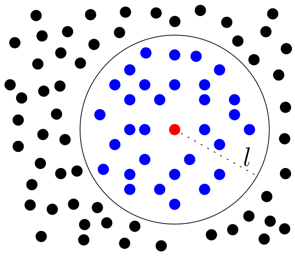


Figure 3.3: Particles enclosed in the circle of radius l are said to be *interacting* with the particle placed at the center.

Evidently, the computational cost of a straightforward calculus of the IL is of order N^2. This would be far from acceptable in cases of a large number of particles. Several advanced techniques can be found in literature to avoid this huge amount of CPU time. Tree search algorithms (Hernquist *et al.* (1989)), Bucket algorithms (Liu *et al.* (2001)) or parallelization techniques (Harada *et al.* (2010)) can be mentioned as examples.

3.3.2 Some further tools in IL calculus

Now, some simple tools in IL calculus can be introduced, saving in this way an appreciable amount of CPU time.

IL storing

In order to establish if a particle is close to a given particle within a disk of radius hk, the Euclidean distance between the particles i and j has to be computed, and this requires exactly six operations:

1. two subtractions: $d_1 = x_i - x_j$, $d_2 = y_i - y_j$
2. two squares: $a = d_1^2$, $b = d_2^2$
3. a sum and a square root: $d = \sqrt{a+b}$

Nevertheless if $|d_1| > hk$ or $|d_2| > hk$ the particle at hand does not need to be stored and the other operations are unnecessary. More precisely it is possible to create a hierarchy of IF blocks in order to prevent every redundant operation, depicted in the flow chart of Figure 3.4

IL refreshing

The technique we are going to describe has been developed in Verlet (1967) and is widely known as *Verlet-List*.

The latter has been used in Panizzo (1994) and in Dominiguez *et al.* (2011) in which several tools are described to improve this technique in order to obtain a more efficient implementation. We shall give an explicit example of its computational advantage.

The underlying idea can be illustrated as follows.

Usually the interaction list is updated at each time-step. Nevertheless, by taking into account the equations of the RK solver (2.30), it is clear that the particles are not moving so much at each time-step, as σ is chosen very small.

Let $l := hk$ be the radius of the compact support of the kernel and $\delta \in \mathbb{R}^+$ suitably

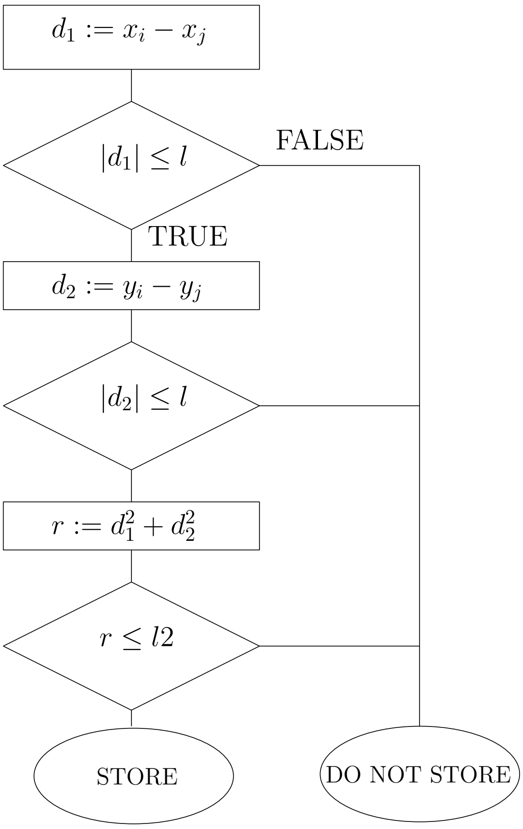


Figure 3.4: The sequence of IF blocks to store particles. The constants $l := hk$ and $l2 := l^2$ are computed once and for all.

chosen. The main idea is to consider an auxiliary structure $\mathcal{I}_\delta$, called *extended IL*, in which particles closer than $l + \delta$ to every single particle are stored. In order to preserve efficiency without loss of information about the particles positions, it is necessary to avoid that a particle $\underline{x}_j$, initially external to $\mathcal{I}_\delta$ (depicted in red in Fig. 3.5), could interact with $\underline{x}_i$, without being stored in the IL. This implies that $\underline{x}_j$ has covered a distance greater than δ.

The maximum displacement at the $q-$th time step, over the particles indexed by i, is bounded by

$$\lambda_q := (\sup_{m=1,\dots,N} |\underline{v}_m|\sigma)_q, \tag{3.42}$$

where $\underline{v}_m$ is the velocity vector of the $m-$th particle of the whole fluid.

Criterion 3.3.1. *Suppose that for* $q = 1$, $\underline{x}_j \notin \mathcal{I}_\delta$. *This implies* $\underline{x}_j \notin \mathcal{B}_l(\underline{x}_i)$ *for all* $q = 1, ..., \tilde{N}$ *such that*

$$\sum_{q=1}^{\tilde{N}} \lambda_q \leq \delta \tag{3.43}$$

This is a necessary condition in order to guarantee that the particle $\underline{x}_j$ is interacting with $\underline{x}_i$ (a straight line displacement). Clearly it is not-sufficient, in the sense that a particle could cover a distance greater or equal to δ without crossing $\partial\mathcal{B}_l(\underline{x}_i)$, as particle $\underline{x}'_j$ depicted in Fig. 3.5 .

Let us discuss in detail the numerical consequence of the above described criterion for the IL refreshing. Let T_r be the time needed to refresh the IL. The data of Table 3.1 arise from one of the most expensive (and accurate) simulations on the simple dam-break problem. It is evident from this table that the number of IL refreshing that we avoid in this way is $96522 - 1931 = 94591$ i.e. the 98%, equivalent to a saved performance time of $94591 * 0.79s = 74726.89s \sim 21h$.

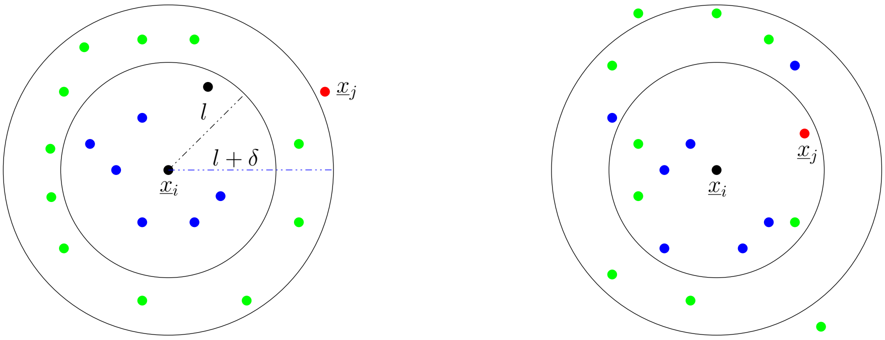


Figure 3.5: The criterion of the IL refreshing. In blue, the set of all interacting particles with $i-$th particle. Green and blue particles are stored in $\mathcal{I}_\delta$. The $j-$th particle is approaching to the particle. On the right the evolved set after condition 3.43 is satisfied: $j-$th particle is actually interacting with the $i-$th and the IL has to be refreshed.

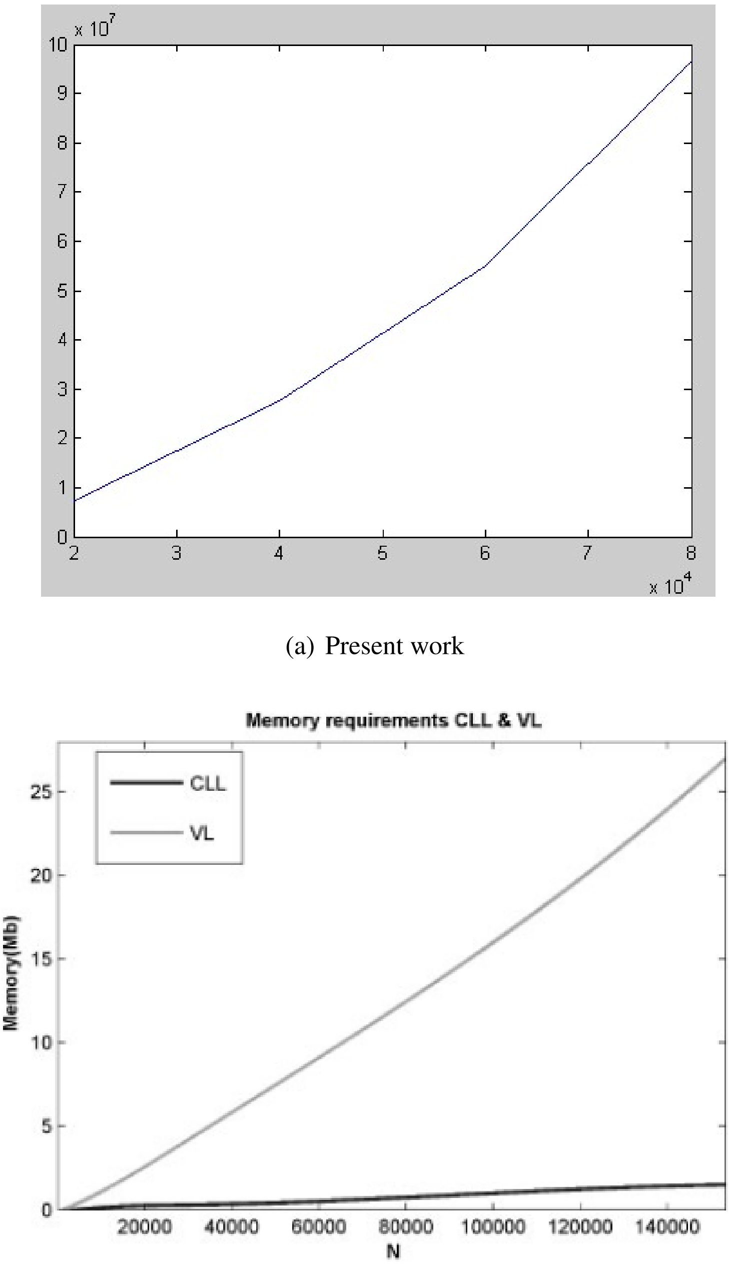
N. of fluid particles	3600
N. of boundary particles	350
Total solver steps	96522
N. of IL refreshing	1931
T_r	0.79s

Table 3.1: Data from simulation of a simple dam-break problem

3.3.3 IL efficiency

In this section we are going to reference and compare computational efficiency data of the above described IL refreshing with the paper Dominiguez *et al.* (2011). In particular, the behaviour of the computational cost in terms of memory and time has been studied as a function of the number of fluid particles and compared with the graphs excerpted from the above mentioned paper.

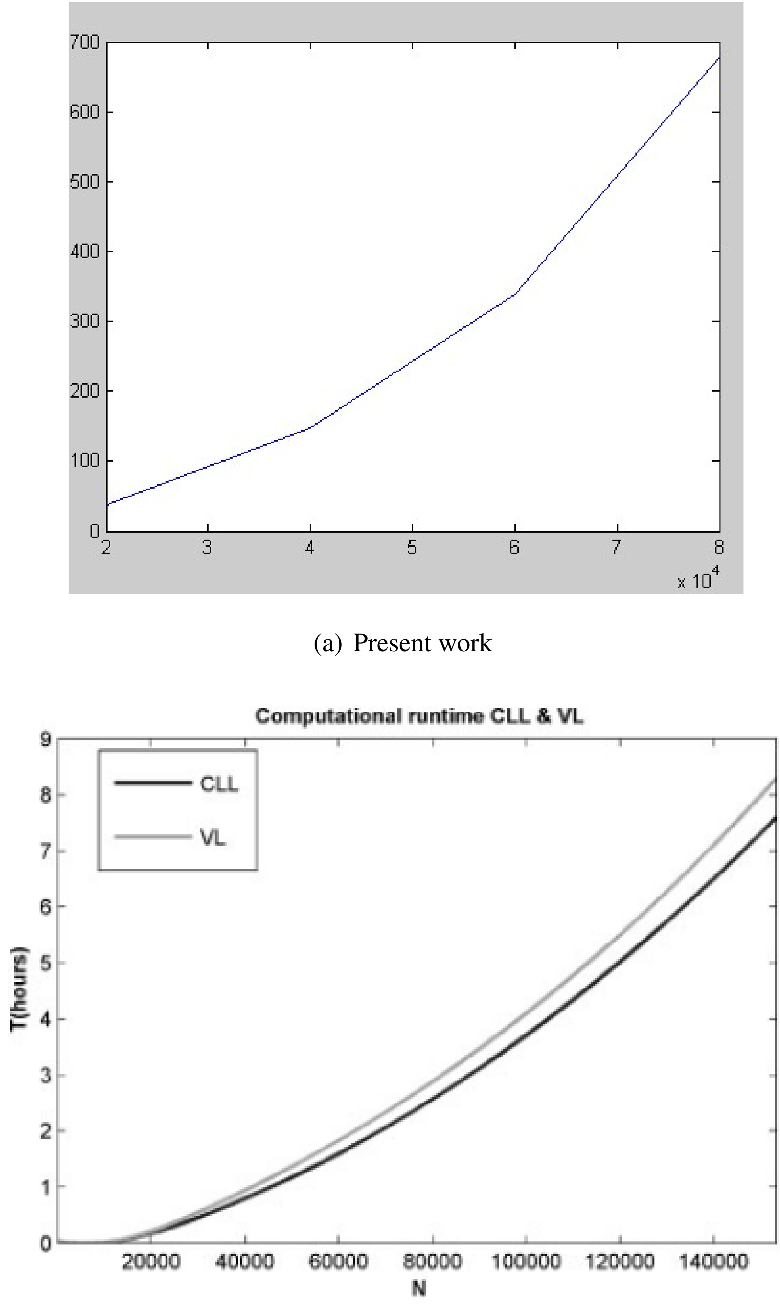
In order to produce these data we considered a set of N randomly distributed particles on a bounded region of the plane (reproducing the most general situation during an SPH simulation). The Figs. 3.6 and 3.7 have been produced by using a range for N from 20000 to 80000. Even if the range used in Dominiguez *et al.* (2011) is larger than $N = 140000$, it is not difficult to recognize the advantages of the IL refreshing presented in this thesis.



(a) Present work

(b) Dominiguez *et al.* (2011)

Figure 3.6: Memory used as a function of N. In subfigure (a) the $x-$axis represents N while the memory occupied, expressed in byte, is on the $y-$axis. In subfigure (b) the same setting is used. It should be remarked that the graph to be compared with (a) is denoted with VL (Verlet List).



(a) Present work

(b) Dominiguez *et al.* (2011)

Figure 3.7: Elapsed time as a function of N. In subfigure (a) the $y-$axis scale is expressed in seconds. The $x-$axis remains unchanged.

In both cases an appreciable difference can be noticed. From Fig. 3.6 is clear that the memory used to store the IL associated to $N = 80000$ particles in the case (a) requires less than 10 Mb, while in the case (b) it is between 10 and 15 Mb.

As for the elapsed time in the same case of $N = 80000$, the difference is much more evident. The required time of nearly 3 hours of Dominiguez *et al.* (2011) is reduced to less than $700s$ i.e. less than 12 minutes.

For a true comparison, each method should be implemented in the same code.

3.4 Dam-break

The dam-break problem consists of the collapse of a water column of a given volume, under the action of gravity, and impacting against a rigid vertical wall. Several variants of this most famous problem can be found in literature as an SPH test case. Monaghan, in Monaghan (1994), used a total number of 2910 particles for the simulation. Recently, many other improvements were obtained, e.g. Armanini *et al.* (2010), where the problem has been extended to $3D$, with the use of different levels of refinement, from 250000 up to 2 million particles and parallelized on 256 CPUs units. In our case, data and results obtained are produced on a single CPU unit laptop. Reasonably, this technological limi-

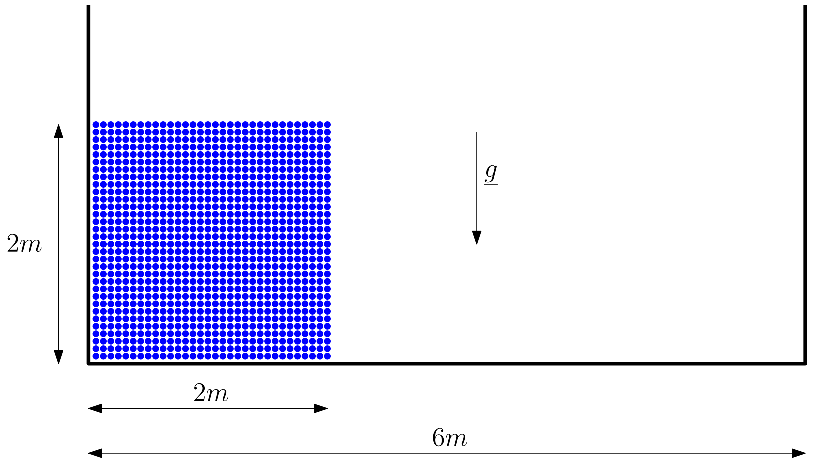


Figure 3.8: The Dam-break setting used in our simulation

tation, pushed and motivated us to optimize the implementation performance avoiding, as much as possible, redundant operations and by using more efficient tools (such as RKF45, our IL refreshing criterion and so on). Of course, as a consequence, this will have a strong impact on the implementation on a powerful machine.

After the simple dam-break problem, the presence of an obstacle will be considered, in order to produce a completely different break-up of the wave generated in the collapse.

Once the preliminary overview of the SPH technique and a discussion of its computational aspects has been completed, it is now possible to address the core of the present work, the SPH formulation of the dam-break, which will be the aim of the next chapter.

The discussion will concern different formulations of the above mentioned problem. In particular, the results related to a different kernels, geometries and number of particles will be presented and compared with existing results in literature. The study on different geometries includes a well known variant of the classical dam-break, obtained by adding a triangular obstacle on the dry bed.

Chapter 4

Implementation

4.1 The boundary conditions problem

One of the main issues when dealing with the SPH formulation is the boundary treatment. A correct understanding of this aspect is a crucial point in order to obtain an appropriate behaviour of the wave impact against the wall on the right.

The simple impenetrability of the boundaries, provided in a wide class of problems, does not have an obvious implementation. In Monaghan (1994), the use of the classical Lennard-Jones potential, arising from molecular dynamics, has been suggested. In this way, the force acting on a particle located at a distance r from the boundary is described by

$$\underline{f}(r) = \frac{1}{r}\left(\left(\frac{r_0}{r}\right)^{p_1} - \left(\frac{r_0}{r}\right)^{p_2}\right)\underline{n}, \qquad p_1 > p_2, \tag{4.1}$$

where $\underline{n}$ denotes the unit vector perpendicular to the boundary.

It is clear that the particles are repelled by a force which modulus tends to infinity as these approach to the boundary. This model may lead to certain instability phenomena, since many ODE solvers could not work properly with unbounded force fields.

A natural way to avoid this kind of problems is to treat a boundary as fixed particles. These interact as standard fluid particles but remain in the same position for all t; with this approach the boundary impenetrability is a mere consequence of the continuity equation, see Crespo *et al.* (2007).

4.2 First SPH formulation

A basic implementation of the problem at hand, fully exhibits a wide range of problems to solve, in order to obtain more accurate results. The initial features of the implementation were:

1. Explicit Euler ODEs solver
2. Straightforward computing of the interaction list
3. Lennard-Jones based boundary conditions.

Since it is a first-order method, the use of an explicit Euler scheme requires a very small time-step and then a very large CPU time, despite the number of particles used was just of approximately 300. It is appropriate to recall that the time-step cannot be chosen to be arbitrary small due to the accumulation of numerical errors at every solver step. For every numerical scheme it is necessary to give a criterion for an appropriate time-step size to choose, in order to prevent either the above mentioned accumulation of errors or discretization errors, related to a too large time-step (see Fig. 4.1). The above mentioned criterion is described in section 2.3.2. As expected, the use of Lennard-Jones force between fluid particles and boundary, leads to an instability of the particles close to the boundary just after few seconds. By looking at formula (4.1) it is clear that, for r approaching to zero, the modulus of the force grows unboundedly due to the singularity at $r = 0$. This implies that the more the fluid particles get close to the boundary (due to pressure of overlying particles), the bigger the repelling force becomes, and then an unbounded growth of the particles acceleration.

As it is well known, explicit Euler method is a Taylor expansion based scheme. For instance, let $\underline{x}(t)$ be the position of a particle, and let $\sigma > 0$ be a small time-step, therefore the position at the time $t + \sigma$ is given by

$$\underline{x}(t+\sigma) = \underline{x}(t) + \dot{\underline{x}}(t)\sigma + \frac{1}{2}\ddot{\underline{x}}(t^*)\sigma^2, \qquad t^* \in (t, t+\sigma), \tag{4.2}$$

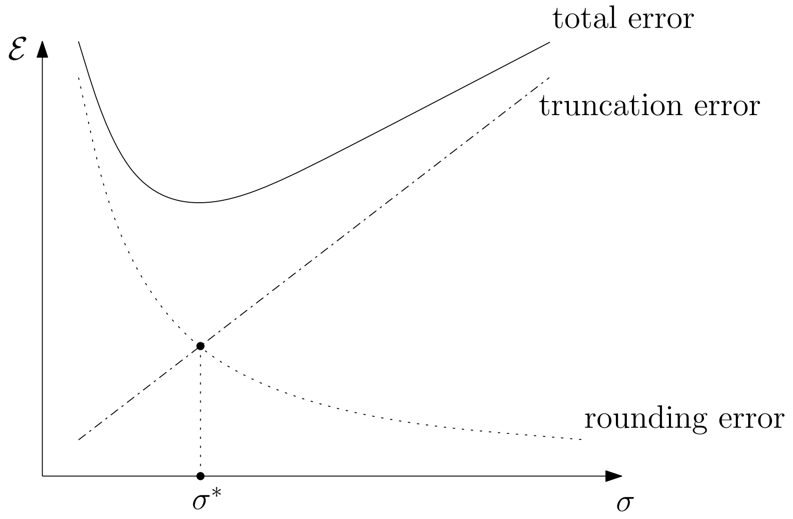


Figure 4.1: Behaviour of the numerical errors $\mathcal{E}$ (in the case of first order Euler method) as a function of the time-step size and its optimal choice σ^*. Figure from Comincioli (2010).

but $\ddot{\underline{x}}$ is proportional to the force acting on the particle, and if the latter is unbounded, the remainder is not small with respect to the first order truncation, implying that the approximation does not make any sense.

This instability phenomenon rapidly propagates through the entire fluid with a complete loss of hydrodynamical reliability almost everywhere, more precisely, the simulation does not describe a physical phenomenon anymore.

4.3 Original SPH formulation

The main features of the developed simulation are:

- RKF45 time integrator,
- Total absence of corrections of non-physical nature (such as artificial viscosity, tensile correction, XSPH variant, etc.),
- Optimal second order kernel constructed in section 3.1.2,
- IL refreshing criterion as described in section 3.3.2,
- Boundary treatment by using ghost particles.
- CSPM formulation of the flow equations

The use of an higher order time scheme (also with respect to the application usually mentioned in literature) and the tools described in section 3.3.2 allowed us to substantially increase the number of fluid particles with respect to the earlier version of the code without drastically affecting the performance time. In the final simulation, whose frames are depicted below, up to 3950 particles were used, with an elapsed time of less than seven hours[1].

As previously mentioned, another key feature of the RKF45 integrator is the adaptivity. This allowed us to use a larger time-step before the impact against the wall as the phase flow is regular and the motion of the particles is ordered. Due to the shock, the latter properties are no longer true and a smaller time-step is required in order to preserve numerical stability and hydrodynamical reliability. The time-step evolution is depicted in Fig. 4.2

[1] On a single CPU laptop machine, Intel ® Celeron ® M Processor 410, 533MHz.

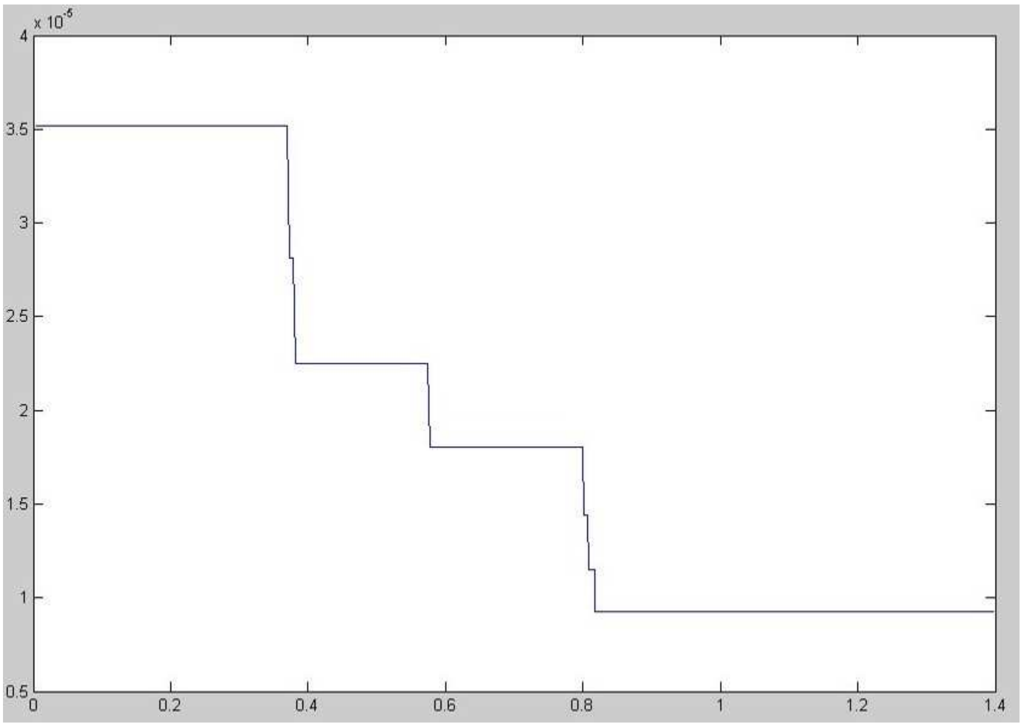
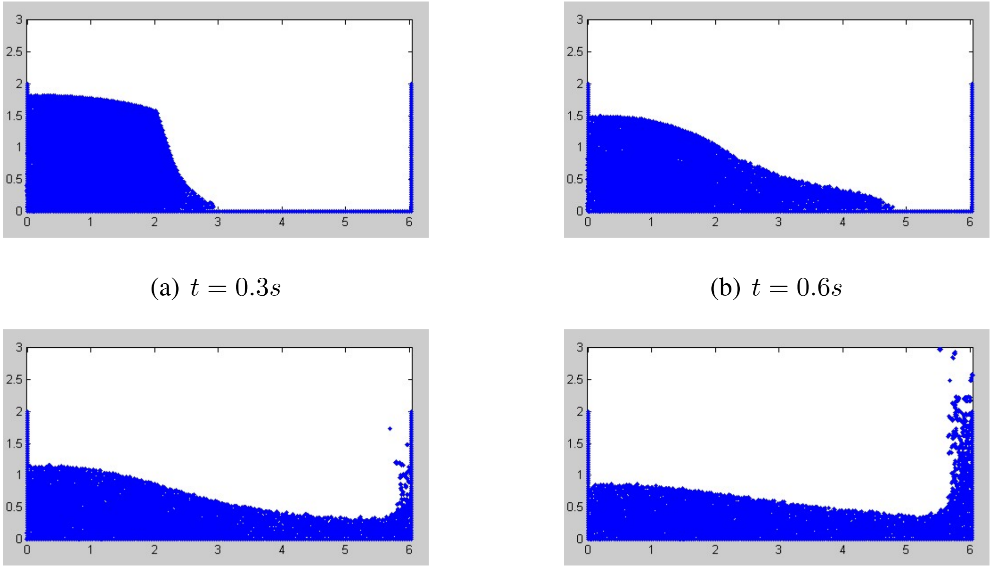


Figure 4.2: The time-step decreasing performed by RKF45. The $x-$axis represents time t (expressed in seconds) and the $y-$axis the time-step size (in seconds) for RKF45. As it can be seen, the minimum value is reached when the fluid mass impacts the right wall (compare with Fig. 4.3(c)).



(a) $t = 0.3s$

(b) $t = 0.6s$

(c) $t = 0.9s$

(d) $t = 1.2s$

Figure 4.3: Four frames from the classical Dam-break simulation at four different instants. The axis scales are expressed in meters. The reported frames have been produced by using a time detection device in the algorithm and their accuracy is $\pm\sigma$ with respect to each t_i, where σ is the current time-step.

4.4 Comparison with existing results

The purpose is now to compare the above described results with an existing simulation in a straightforward way. The simulations reported in Gomez-Gesteira *et al.* (2010) have been chosen as a test bench. For a more adequate comparison, the geometry of the discussed implementation has been adapted to the setting used in Gomez-Gesteira *et al.* (2010), see Fig. 4.4.

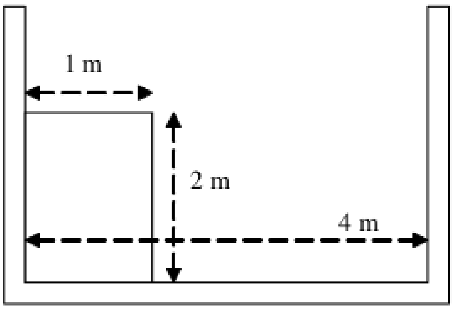


Figure 4.4: Initial configuration of the water column and the tank used in Gomez-Gesteira *et al.* (2010).

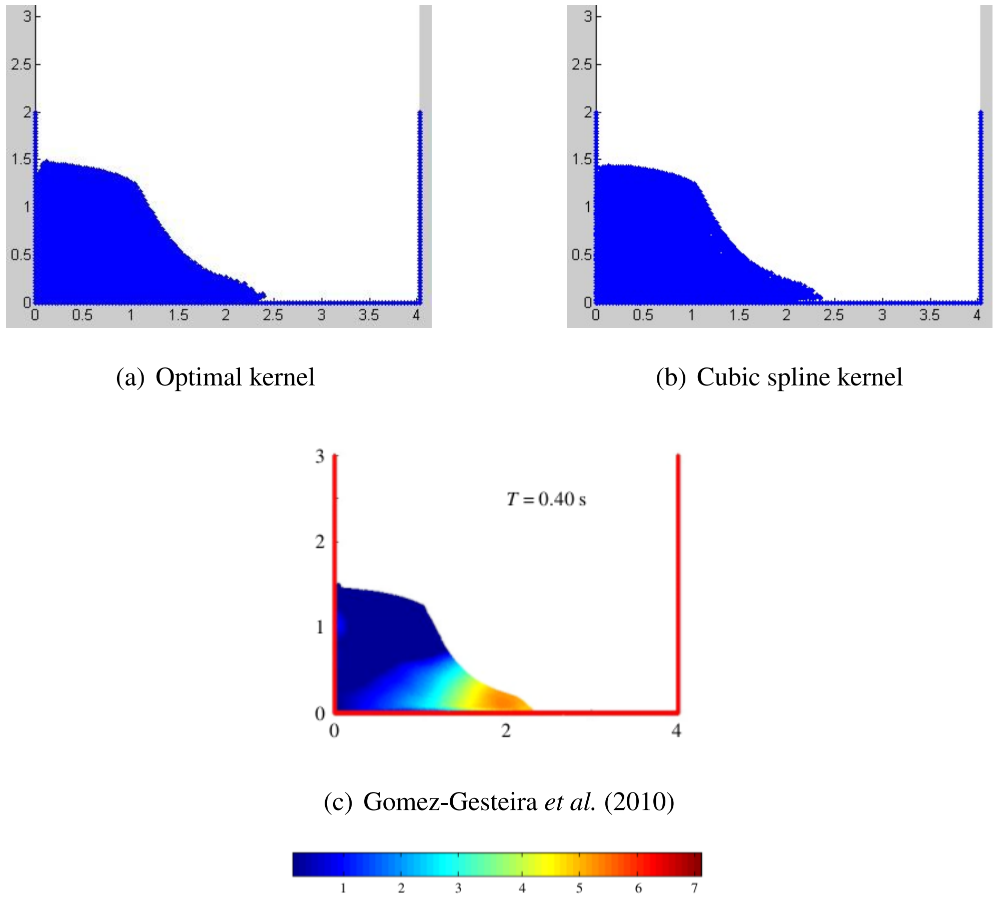
The implementation used in the present section has been performed with 3200 fluid particles and 282 boundary particles, as a consequence the smoothing length has been chosen as $h = 0.03m$. Either the Optimal kernel (constructed in Sec 3.1.2) and the classical Cubic spline kernel (see 3.36) have been used.

This section is divided in two parts. The first is devoted to the analysis of the two water columns positions, sampled at the same instants, more precisely, $t_i := \{0.4, 0.6, 0.8, 1\}$s, determined with the same criterion described in Fig. 4.3.

In the subsequent part the analysis of the position of two wave toes and the height of the collapsing dam are going to be discussed.

4.4.1 Position of the water column

First frame



(a) Optimal kernel

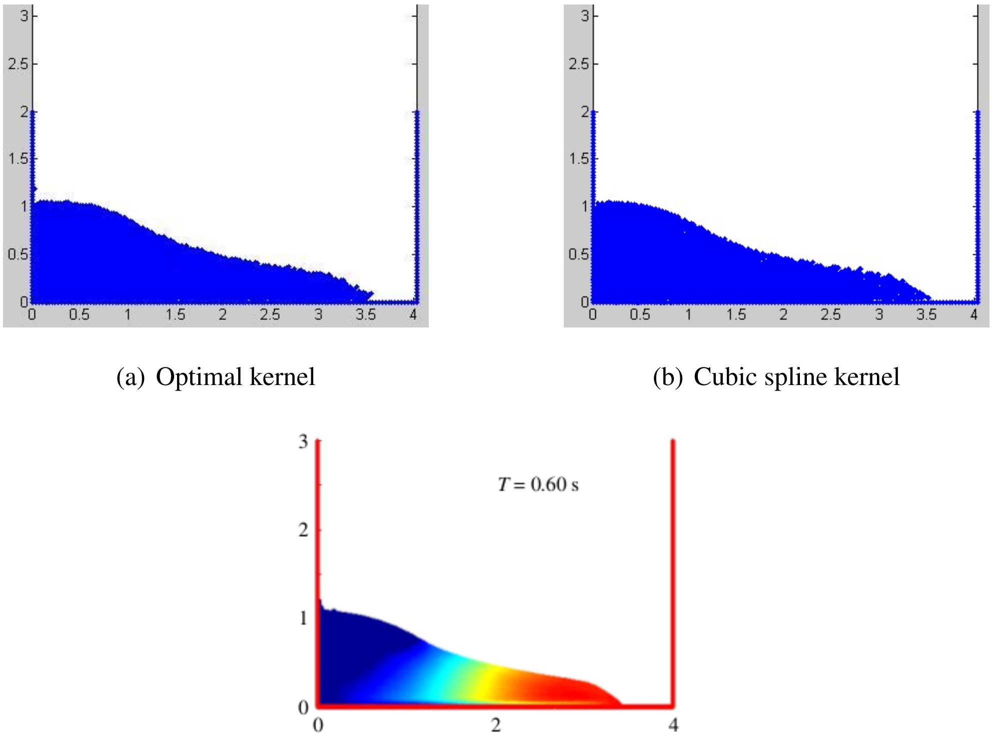
(b) Cubic spline kernel

(c) Gomez-Gesteira *et al.* (2010)

(d)

Figure 4.5: Frames from the dam-break simulation at $t = 0.4$s. Fig. 4.5(a) and 4.5(b): frames from the simulations performed by using Optimal and Cubic spline kernel, respectively. Fig. 4.5(c) represents the position of the water column as excerpted from Gomez-Gesteira *et al.* (2010). The axis scales are expressed in meters. Fig. 4.5(d) expresses particle velocities of 4.5(c) (in m/s). The subsequent figures are similarly organized.

Second frame



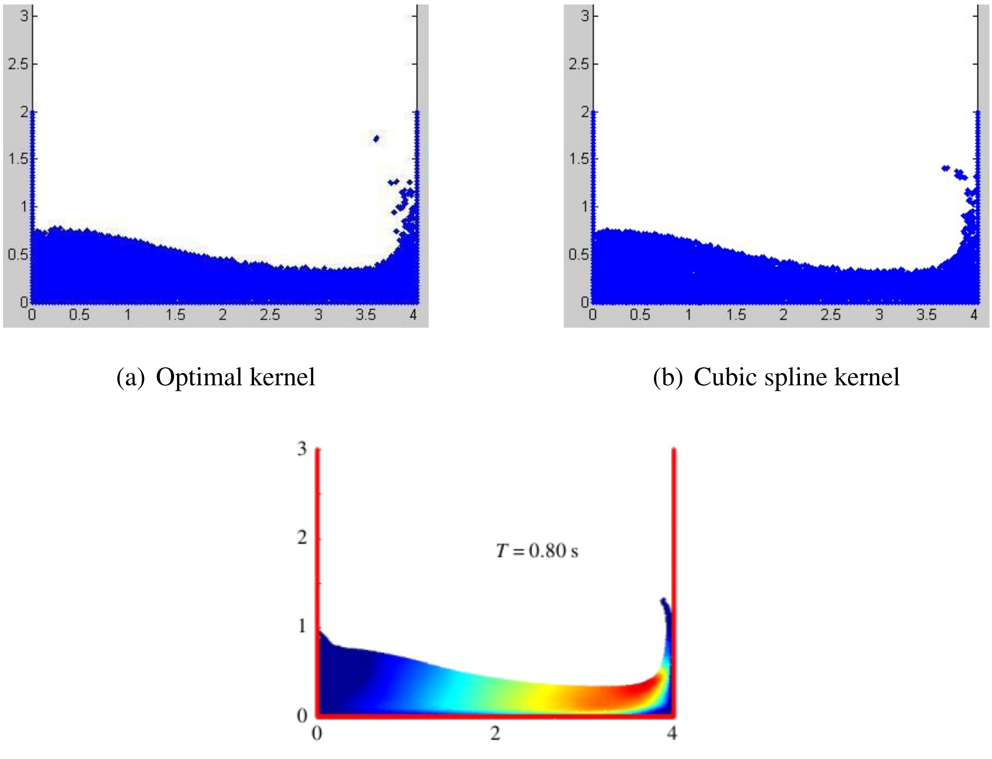
(a) Optimal kernel

(b) Cubic spline kernel

(c) Gomez-Gesteira *et al.* (2010)

Figure 4.6: Frames from the Dam-break simulation at $t = 0.6$s.

Third frame



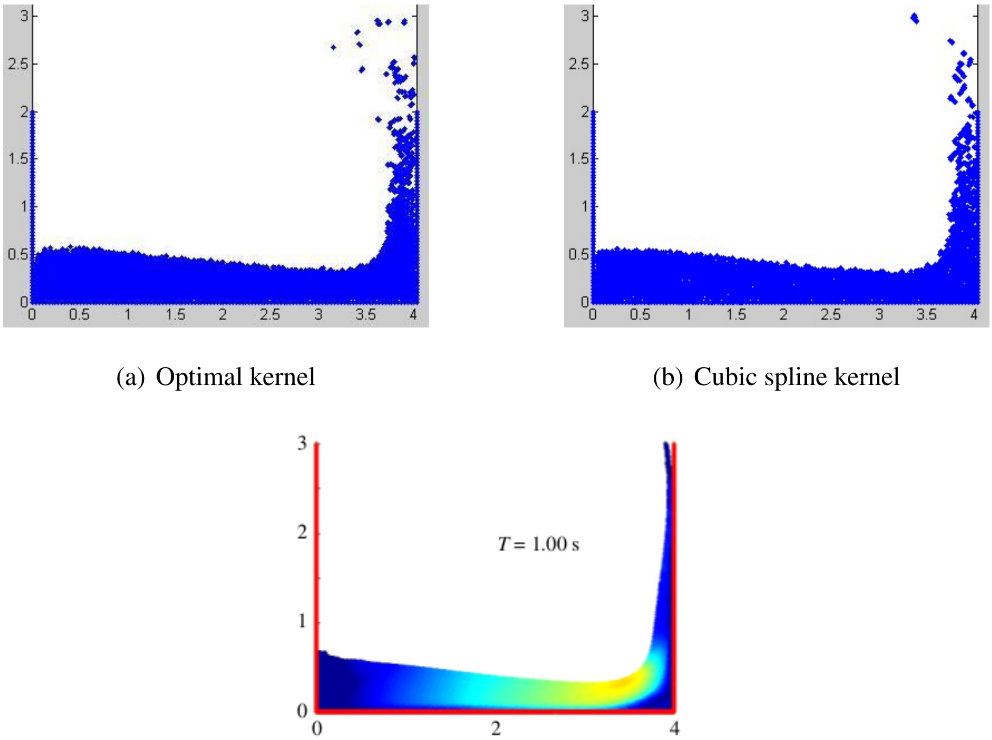
(a) Optimal kernel

(b) Cubic spline kernel

(c) Gomez-Gesteira *et al.* (2010)

Figure 4.7: Frames from the Dam-break simulation at $t = 0.8$s.

Fourth frame



(a) Optimal kernel

(b) Cubic spline kernel

(c) Gomez-Gesteira *et al.* (2010)

Figure 4.8: Frames from the Dam-break simulation at $t = 1$s.

As can be easily seen from the four frames a full similarity between the simulations performed by using the two different kernels can be appreciated. Despite some differences due to the absence of post formulation corrections, both of the wave profiles occupy a position that is fully comparable to the frames excerpted from Gomez-Gesteira *et al.* (2010).

4.4.2 Evolution of the toe and height

A further tool to analyse the similarity with the chosen reference consists in comparing the time evolutions of the dam toe and height.

Position of the toe

The consistency of the simulations performed by using the two different kernels (Optimal and Cubic spline) are deduced by a straightforward comparison in the Figs. 4.9 and 4.10 below. These represent the position of the toe of the collapsing dam as a function of time (continuous line) and a set of data as provided by Koshizuka *et al.* (1996) (dots), taken from Gomez-Gesteira *et al.* (2010).

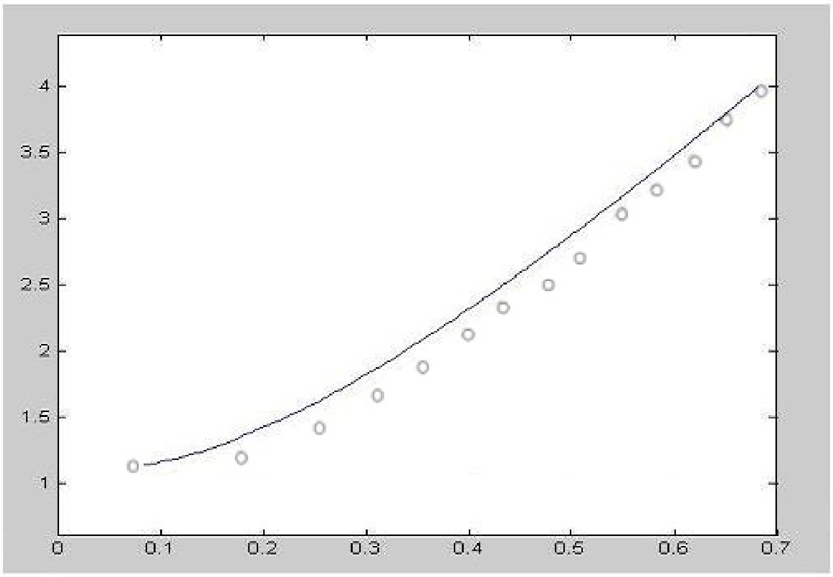


Figure 4.9: Position of the toe in the Optimal kernel case (continuous line) and experimental data from Koshizuka *et al.* (1996) (dots). The $x-$axis scale is expressed in seconds, while the $y-$axis scale is expressed in meters.

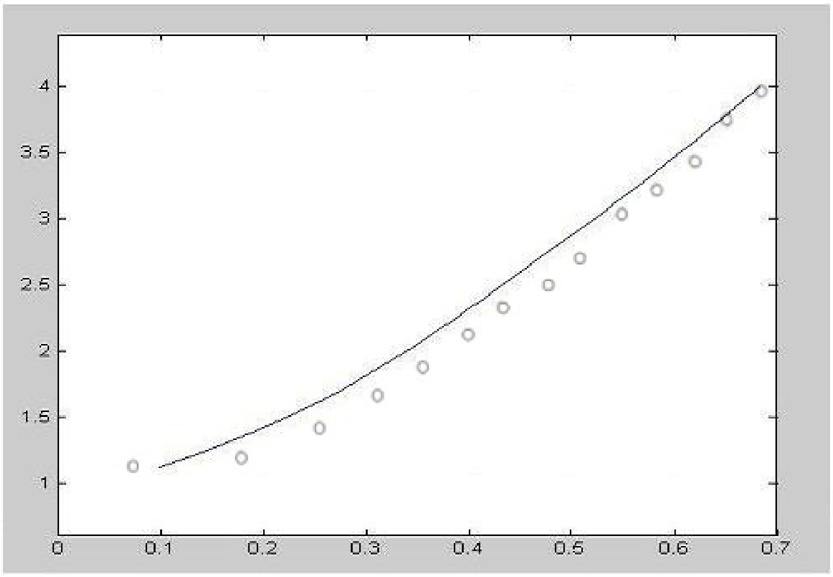


Figure 4.10: Position of the toe in the Cubic kernel case (continuous line) and experimental data from Koshizuka *et al.* (1996) (dots). The axes scales are organized as in Fig. 4.9.

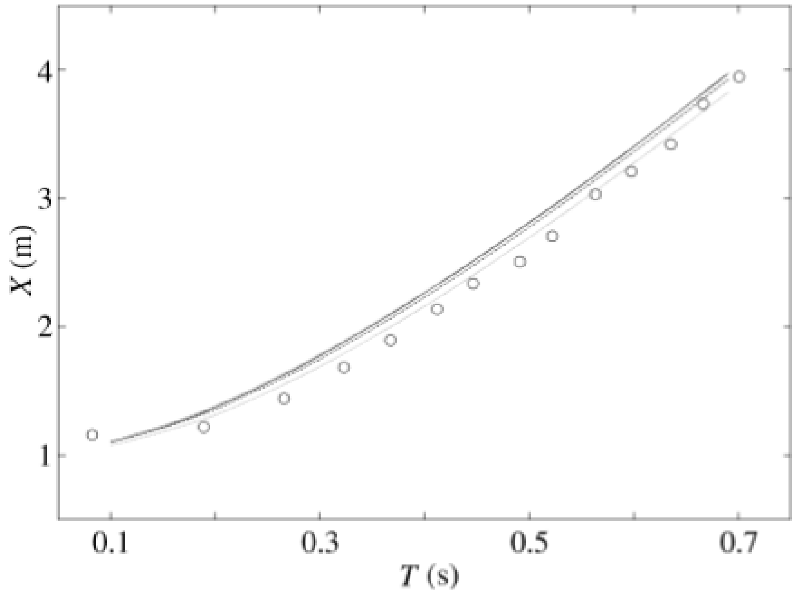
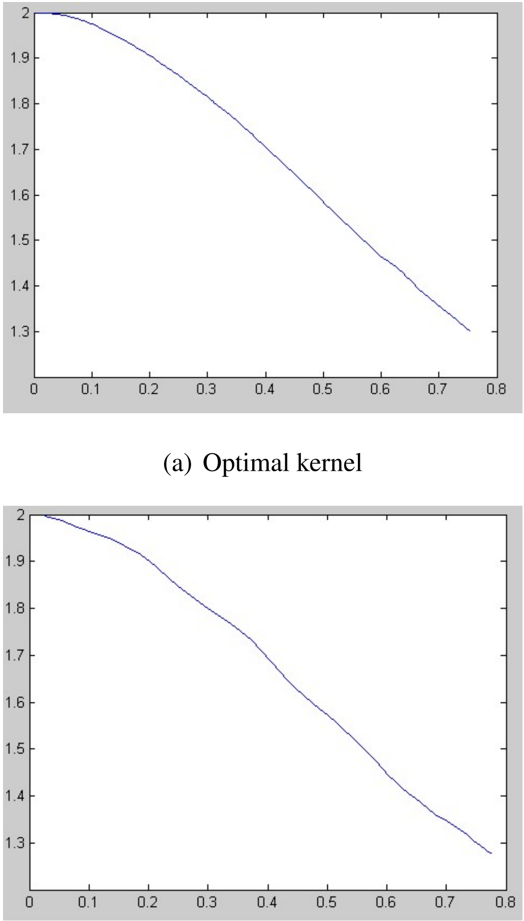


Figure 4.11: Position of the toe as excerpted from Gomez-Gesteira *et al.* (2010). The three curves correspond to different numerical discretization, more precisely for h chosen as $0.0390m$ (top curve), $0.0260m$ and $0.0156m$, respectively. Dots represent experimental data provided by Koshizuka *et al.* (1996).

Height of the wave

In order to compare the simulations which toe evolutions have been described by Figs. 4.9 and 4.10, the analysis can be completed by reporting the corresponding plots for the heights of the collapsing dams.



(a) Optimal kernel

(b) Cubic spline kernel

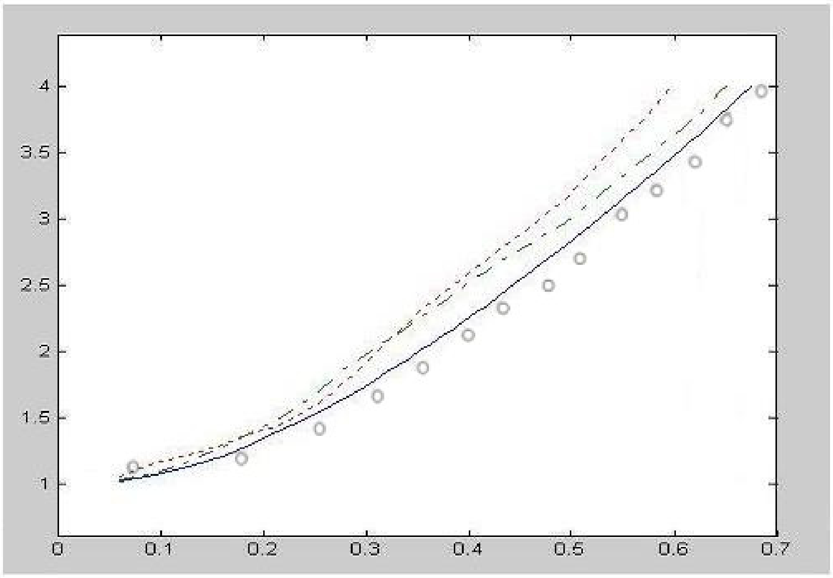
Figure 4.12: Evolution of the highest point of the collapsing dam as a function of time (reported on the $x-$axis and expressed in seconds). The $y-$axis scale is expressed in meters.

Despite the fact that the two graphs are fully comparable, in the Optimal kernel case it is possible to appreciate a smoother evolution of the height.

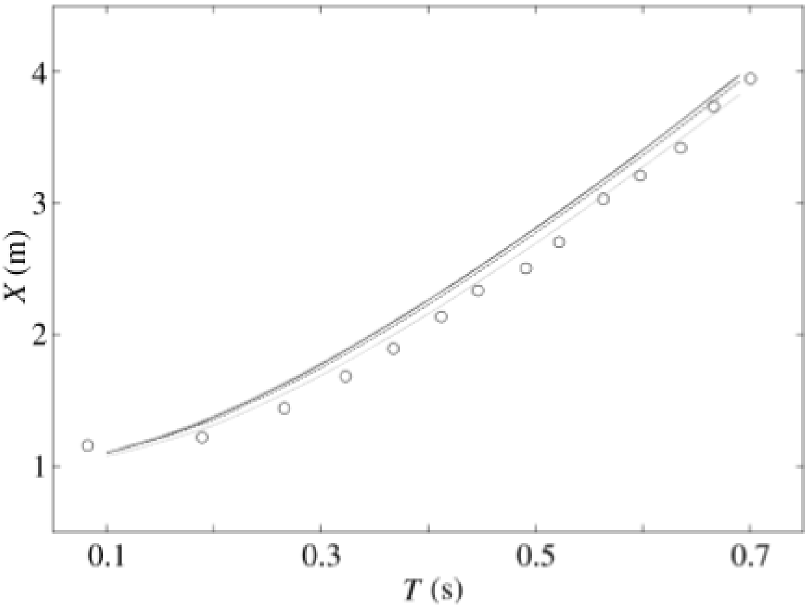
4.4.3 Qualitative convergence study

In this section the behaviour of the implementation is studied in case of coarser particles. The toe evolutions corresponding to three different values are reported in Fig. 4.13, together with a straightforward comparison with Gomez-Gesteira *et al.* (2010).

As can be easily deduced by Fig. 4.13 as the particles size decreases, the trajectory of the toe becomes smoother and closer to the trajectory reported by Gomez-Gesteira *et al.* (2010) and to the experimental data.



(a)



(b)

Figure 4.13: In (a), position of the toe for the following values of the smoothing length h: $0.067m$ (900 particles, dotted line), $0.05m$ (1434 particles, dash dotted line), $0.04m$ (2112 particles, continuous line) and experimental data excerpted from Koshizuka *et al.* (1996) (dots). Fig. (b) excerpted from Gomez-Gesteira *et al.* (2010) for h chosen as $0.0390m$ (top curve), $0.0260m$ and $0.0156m$, respectively. Dots represent experimental data provided by Koshizuka *et al.* (1996).

4.5 Dam-break with an obstacle

The aim is now to test the implementation on a very interesting and widely-used variant of the classical dam-break problem, see e.g. Liu *et al.* (2003). In this case a small obstacle with the shape of a right-angled isosceles triangle is positioned on the ground.

Until the wave reaches the obstacle, the problem is fully analogous to the classical dam-break; however the break-up after the impact is substantially different from the previous setting. In this case, the right wall has been removed in order to study the wave produced by the impact. This problem should be regarded as an indicative model of a violent impact against a fixed marine structure such as platforms or coastal defences.

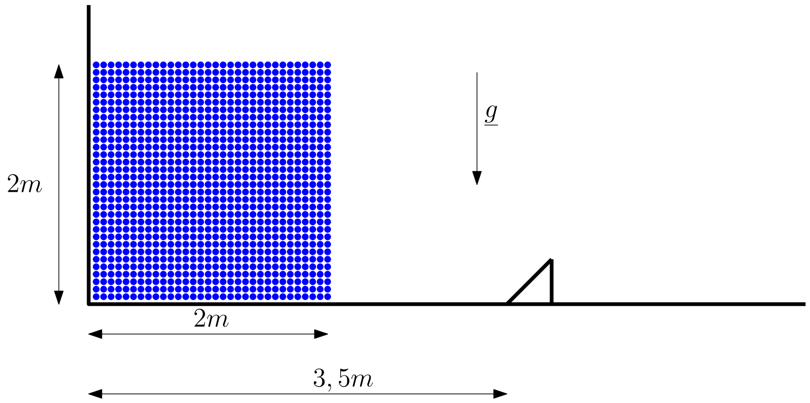
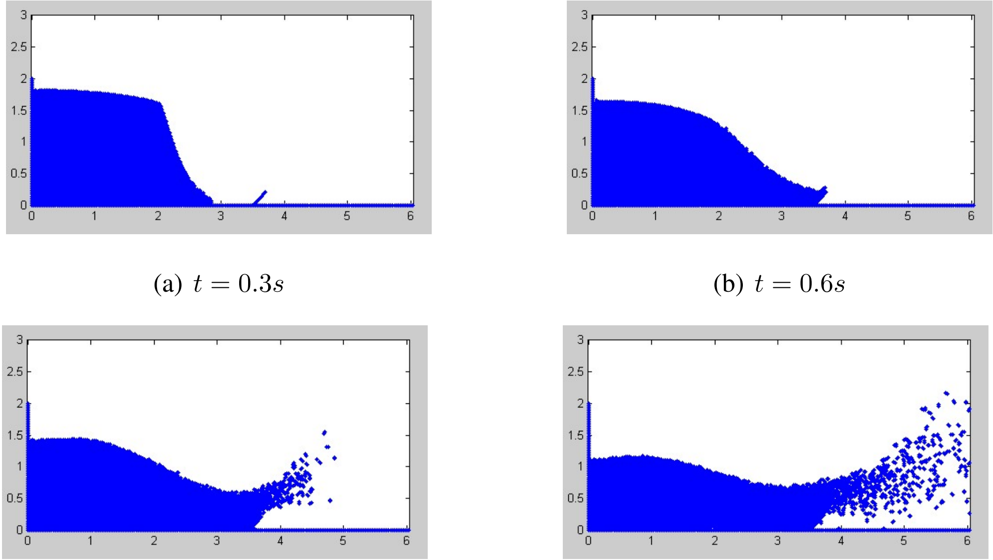


Figure 4.14: The dam-break with a triangular obstacle. The equal sides have been chosen 40cm long.

The simulation has been done with exactly the same original SPH formulation, through a simple modification of the boundaries configuration. The simulation time is slightly smaller (equal to $1s$) than the classical problem as it is not needed to examine the effect due to the right wall on the wave.



(a) $t = 0.3s$

(b) $t = 0.6s$

(c) $t = 0.9s$

(d) $t = 1.2s$

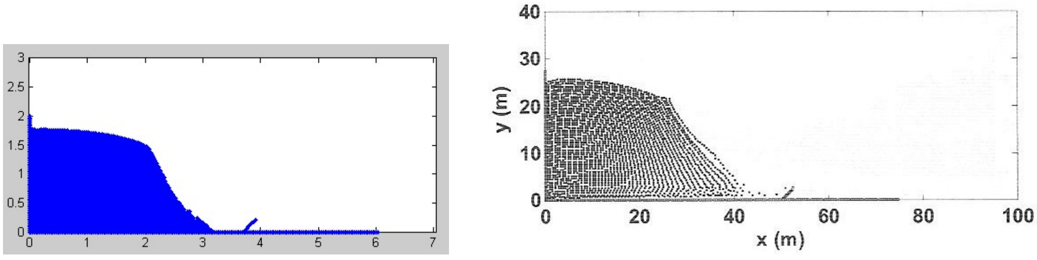
Figure 4.15: Four frames from the dam-break simulation with an obstacle; the initial condition is the same as depicted in Fig. 4.3(a). The axis scales are expressed in meters.

4.5.1 Comparison

As for the classical case, a direct comparison with an existing example is performed. In such case, the frames depicted below, and labelled with (b), have been excerpted from Liu *et al.* (2003). The frames labelled with (a), corresponding to the formulation described in Sec. 4.5, have been computed using a wave front detection control and taking into account of the different scales used by the two formulations. The total number of particle used is 3919.

An appreciable similarity between the two sets of frames can be observed. As discussed in the classical dam-break case, the obtained wave profile looks more noisy than (b) (from Liu *et al.* (2003)), due to the complete absence of artificial corrections. Nevertheless, from the physical point of view, it is not completely clear which one is more reliable: in cases (b), any kind of particle fragmentation is completely absent, whereas it should naturally be produced because of the wave impacting against the obstacle.

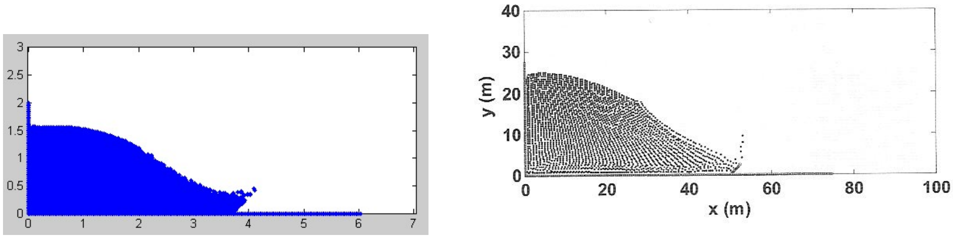
First frame



(a) (b)

Figure 4.16: Comparison between the Dam-break variant simulations, subfigure (b) is excerpted from Liu *et al.* (2003) for $x = 3.11m$.

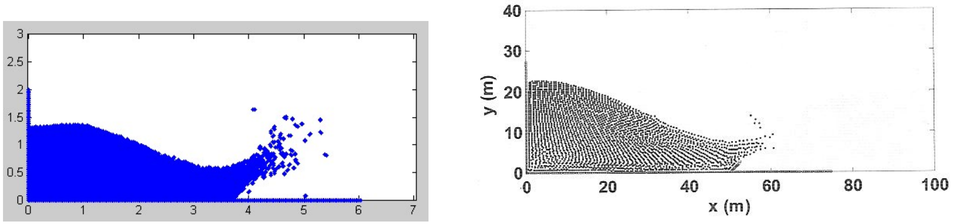
Second frame



(a) (b)

Figure 4.17: Comparison between the Dam-break variant simulations at $x = 3.91m$ (left) and Liu *et al.* (2003) results (right).

Third frame



(a) (b)

Figure 4.18: Comparison between the Dam-break variant simulations at $x = 4.44m$ (left) and Liu *et al.* (2003) results (right).

Fourth frame

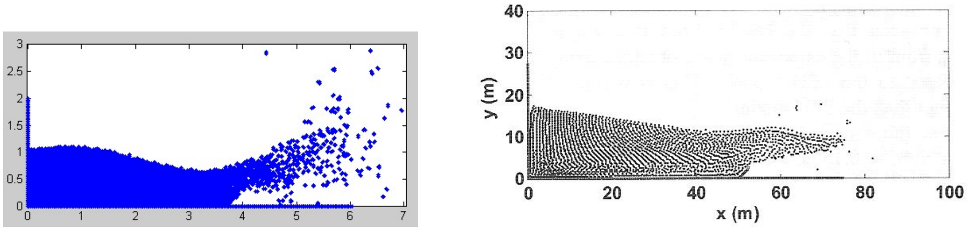


Figure 4.19: Comparison between the Dam-break variant simulations at $x = 5.59m$ (left) and Liu *et al.* (2003) results (right).

The above presented results show the comparability between the SPH formulation described in this work and other relevant results found in literature. The comparative study performed on the optimal kernel, with respect to the well known cubic spline, exhibited an appreciable equivalence between the two interpolation functions.

A summary of the results obtained will be presented in the following chapter. Furthermore, the optimal kernel will be compared with the other standard kernels used in literature, showing that it allows to attain the minimum $O(h^2)$ error.

Chapter 5

Conclusions

5.1 Discussion of results

The main goal of this research was to approach the SPH implementation of dam-break problem differently, to give a critical evaluation of the already existing theory.

The crucial point of the analysis concerned not using of corrections in the fluid equations. It is well known that corrections such as artificial viscosity, XSPH variant, tensile correction and so on, are used mainly to guarantee a better particles behaviour. Basically, the use of additional terms, is a way to force particles to move according to certain laws (more orderly, less noisy), which do not arise from the equations of motion[1]. In this way one gets a description (and then a simulation) of a phenomenon that is not a fluid motion anymore.

On the other hand, it is known that a straightforward SPH approximation, can lead to severe errors in certain situations such as a small number of particles within the support domain and/or not uniformly distributed particles. Nevertheless, these potential errors do not justify the use of the previously mentioned non-physical corrections.

The attention was focused instead, on the SPH discretization accuracy (mainly on theory

[1] In addition, a large number of parameters appearing in such adjustments has to be determined empirically.

of kernels), more accurate ODEs solvers and interaction list storing optimization. As said in the foreword, the original results of this work are centred on these topics.

In particular, the aim of the theory of kernel developed in this thesis, was to construct a kernel function with the following features:

- Does not require the use of tensile correction as its second derivative is positive on the support domain by construction;
- Leads to an error of order h^2 unavoidable for non-negative (i.e. non-physical) function, times a constant with the smallest value achievable.

This theory can be generalized in principle to arbitrary high degree or piecewise kernels, even if this approach requires to deal with a non-linear system that has to be solved, in general, with a numerical method.

The use of RKF45 method is a way to obtain either efficiency or accuracy. The adaptive time-step adjustment provided by this scheme, allows to fix a given precision *a priori* (controlling in this way the error throughout the simulation), but can also accelerate the simulation process increasing the time-step, if the chosen one is too small. This is a crucial point as this class of implementations requires, in general, a very large computation time. RKF45 is advantageous with respect to other time marchers frequently used in literature. The widely used Verlet scheme, despite its symplectic feature, it is just a second order method. RKF45, instead, exhibits an error at least of order four, allowing us to choose a time-step hundreds time greater than the Verlet scheme, in order to obtain the same accuracy.

The IL refreshing criterion 3.3.1, has been another decisive tool used to save a large amount of CPU time. The idea to update the IL only if the maximum displacement of the particles (evaluated all over the fluid) is greater than a prefixed quantity, allows to avoid redundant storage processes in case of slow variations due to small time-steps. The results obtained by a joint action of the above mentioned tools are described in Table 3.1.

5.2 Future work

There are several refinement to be taken into account in order to improve the implementation. From the theoretical point of view, a development of the theory of optimal kernels and its application to higher degree kernels and piecewise kernels is highly desirable. Even if, a general treatment (as mentioned in Chapter 3), requires the use of a numerical method to solve the $n-$ dimensional optimization problem.

Another issue to be addressed is the computational efficiency. Mainly, a considerable improvement could be a more efficient interaction list storing algorithms, such as bucket, tree search etc. Such algorithms give rise to a logarithmic computational costs versus the linear cost of a straightforward search. On the other hand, the usage of higher order RK adaptive schemes and their symplectic variants could lead to a further improvement of the accuracy. As it is well known, the symplectic feature of a numerical integrator guarantees the best performance from the energy conservation point of view.

As matter of fact, in the model some adjustments are required: the use of Euler formulation is a fundamental step in a critical analysis but it does not lead to some relevant physical phenomena such as the so-called aerated impacts, bubble entrapment and so on. For this reason the use of multiphase formulation of the fluid equation could be the next step to be achieved.

An interesting problem, actually more closely related to an engineering approach, could be the measurement of the loads exerted in the impact either on the right wall (classical dam-break) or on the obstacle (dam-break with an obstacle) Lobovsky *et al.* (2014).

Appendix A

A second order Taylor formula

In this appendix a detailed proof of the proposition 3.1.2 is given.

Let us choose, without loss of generality $\underline{x}_0 = \underline{0}$. As usual, denoting with $\underline{x} = (x^1, x^2)$, follows

$$\begin{aligned} \int_\Omega f(\underline{x})W(\underline{x})d\underline{x} &= f(\underline{0})\int_\Omega W(\underline{x})d\underline{x} \\ &+ \sum_{i=1}^{2} \frac{\partial f}{\partial x^i}\int_\Omega W(\underline{x})x^i d\underline{x} \\ &+ \sum_{i,j=1}^{2} \frac{\partial^2 f}{\partial x^i \partial x^j}\int_\Omega W(\underline{x})x^i x^j d\underline{x} + o(h^2). \end{aligned} \tag{A.1}$$

The first integral appearing in the right hand side of the previous expansion is clearly equal to one by unity condition. Due to the rotational symmetry of the kernel is convenient to evaluate the other integrals by using polar coordinates:

$$\begin{cases} x^1 = \rho\cos\theta \\ x^2 = \rho\sin\theta \end{cases} \tag{A.2}$$

so

$$\begin{aligned} \Omega &= \{(x^1, x^2) \in \mathbb{R}^2 : \sqrt{(x^1)^2 + (x^2)^2} \le l\} \\ &= \{(\rho, \theta) \in \mathbb{R} \times \mathbb{S} : \rho \in [0, l]\} \end{aligned} \tag{A.3}$$

One gets

$$\int_\Omega W(\underline{x})x^i d\underline{x} = I_i \int_0^l W(\rho)\rho^2 d\rho = 0 \tag{A.4}$$

where $I_1 := \int_0^{2\pi} \cos\theta d\theta = 0 = \int_0^{2\pi} \sin\theta d\theta =: I_2$. While

$$\begin{aligned} \int_\Omega W(\underline{x}) x^1 x^2 d\underline{x} &= \left(\frac{1}{2}\int_0^{2\pi} \sin 2\theta d\theta\right)\left(\int_0^l W(\rho)\rho^3 d\rho\right) = 0 \\ \int_\Omega W(\underline{x})(x^i)^2 d\underline{x} &= \tilde{I}_i \int_0^l W(\rho)\rho^3 d\rho \end{aligned} \tag{A.5}$$

with $\tilde{I}_1 := \int_0^{2\pi} \cos^2\theta d\theta = \pi = \int_0^{2\pi} \sin^2\theta d\theta =: \tilde{I}_2$.

The substitution of the obtained values in the first formula closes the proof.

References

Adams, B. and Wicke, M. (2009). *Meshless Approximation Methods and Applications in Physics Based Modeling and Animation.*, The Eurographics Association.

Anderson, J. D. (1995). *Computational Fluid Dynamics: the basics with applications.*, Mc Graw-Hill.

Armanini, A., Ferrari, A., Dumbser, M., Toro, E. F. *A new $3D$ parallel SPH scheme for free-surface flow.*, 3rd ERCOFTAC SPHERIC workshop on SPH applications. June, 4th - 6th 2008, Lausanne, Switzerland.

Armanini, A., F., Ferrari, A., Fraccarollo, L., Dumbser, M. and Toro, E. (2010). *Three-dimensional flow evolution after a dam-break.*, Journal of Fluid Mechanics, $663 : 456-477$.

Belytschko T., Krongauz Y., Organ D., Fleming M. and Krysl, P. (1996) *Meshless methods: An overview and recent developments.*, Comput. Methods Appl. Mech. Eng. 139, $3-48$.

Benson, D. J. (1992). *Computational methods in Lagrangian and Eulerian hydrocodes.*, Comp. Meth. Appl. Mech. and Eng. $99 : 235-394$.

Bullock, G. N., Obhrai, C., Peregrine, D. H. and Bredmose, H. *Violent breaking wave impacts. Part 1: results from large scale regular wave tests on vertical and sloping walls.*, Coastal Engineering. $54(8) : 602-617$.

Causon, D. M., Mingham, C. G., Ingram, D. M. and Qian, L. (2006). *A free surface capturing method for two fluid flows with moving bodies.*, Proceedings of the Royal Society of London: a, $462(2065) : 21 - 42$.

Chen, J.S., Pan, C. and Wu, C.T. (1997). *Large deformation analysis of rubber based on a reproducing kernel particle method.*, Computational Mechanics $19, 153 - 168$.

Chen, J.S., Pan, C., Wu, C.T., and Roque, C. (1998). *A Lagrangian reproducing kernel particle method for metal forming analysis.*, Computational Mechanics $21, 289 - 307$.

Comincioli, V. *Matematica Numerica.*, Mc Graw-Hill 2010.

Crespo, A. J. C. (2008). *Application of the Smoothed Particle Hydrodynamics model SPHysics to free-surface hydrodynamics.*, Ph.D. Thesis, Universidade de Vigo.

Crespo, A.J.C., Gomez-Gesteira, M., Dalrymple, R.A. (2007). *Boundary conditions generated by dynamic particles in SPH methods*, CMC-Computers Materials and Continua, Vol: 5, Issue 3:, Pagg $173 - 184$.

Cummins, S. J., Silvester, T. B. and Cleary, P. W. (2012). *Three-dimensional wave impact on a rigid structure using smoothed particle hydrodynamics.*, Int. J. Num. Meth. Fluids $69(9) : 1566$.

Dalrymple, R. A. and Knio, O. (2001). *SPH modelling of water waves.*, In Proceedings of Coastal Dynamics.

Dolbow, J., Mose, N., and Belytschko, T. *Discontinuous enrichment in finite elements with a partition of unity method.*, Finite Elem. Anal. Design $36, 235 - 260$.

Dominiguez, J.M., Crespo, A.J.C., Gomez-Gesteira, M. and Marongiu, J.C. (2011). *Neighbour lists in smoothed particle hydrodynamics.*, International Journal for Numerical Methods in Fluids. $67 : 2026 - 2042$.

Durisen, R. H., Gingold, R. A. and Boss, A. P. (1986). *Dynamic Fission Instabilities in Rapidly Rotating $n = 3/2$ Polytropes: A Comparison of Results from Finite-difference and Smoothed Particle Hydrodynamics Codes.*, Astron. J., $305 : 281 - 308$.

Evrard, A. E. (1988). *Beyond $N-$body: $3D$ cosmological gas dynamics.*, Mon. Not. R. Astr. Soc. 235, $911 - 934$.

Garcia-Senz, D., Bravo, E. and Woosley, S.E. (1999). *Single and multiple detonations in white dwarfs.*, Astron. Astrophys. 349, $177 - 188$.

Gingold, R. A. and Monaghan, J. J. (1977). *Smoothed Particle Hydrodynamics: theory and application to non-spherical stars.*, Mon. Not. Royal Astronomical Society 181, $375 - 389$.

Gomez-Gesteira, M., Rogers, B.D., Dalrymple, R.A. and Crespo, A.J.C., (2010). *State-of-the-art of classical SPH for free-surface flows.*, Journal of Hydraulic Research, 48:S1, $6 - 27$.

Hallquist, J. O. (1988). *User's manual for DYNA2D - an explicit 2-dimensional hydrodynamic finite element code with interactive rezoning and grafic display.*, Lawrence Livermore National Laboratory Report UCID - 18756, rev.3.

Hallquist, J. O. (1998). *LS-DYNA THEORETICAL MANUAL.*, Livermore Software Technology Corporation, 2876 Waverley Way, Livermore, California 94550-1640

Hans, U. M. (1999) Review: *Hydrocodes for structure response to underwater explosions.*, Shock and Vibr. $6(2) : 81 - 96$.

Harada, T., Koshizuka, S. and Kawaguchi, Y. (2010). *Smoothed Particle Hydrodynamics on GPUs.*, preprint.

Hernquist, L., Katz, N. (1989). *TreeSPH- A unification of SPH with the hierarchical tree method.*, The Astrophysical Journal Supplement Series, $70 : 419 - 446$

Hirsch, C. (1988). *Numerical computation of internal and external flows.*, Vol. 1, Wiley-Interscience Publication.

Hughes, J. P., Graham, D. I. and Reeve, D. E. *Wave impact simulations using incompressible and weakly-compressible SPH models.*, 3^{rd} ERCOFTAC SPHERIC workshop on SPH applications, Lausanne, June $4-6$th 2008.

Hull, P., Muller, G. (2002), *An investigation of breaker heights, shapes and pressures*, Ocean Eng. $29(1) : 59-79$.

Ivings, M., Causon, D. M. and Toro, E. F. (1998). *On Riemann solvers for compressible liquids.*, International Journal for Numerical Methods in Fluids. $28 : 395-418$.

Koshizuka, S. and Oka, Y. (1996). *Moving-particle semi-implicit method for fragmentation of compressible fluid.*, Nuclear Sci. Engrg. 123, $421-434$.

Lattanzio, J. C., Monaghan, J. J., Pongracic, H. and Schwartz, M. P. (1986). *Controlling penetration.*, SIAM Journal of Scientific and Statistical Computing, $7(2) : 591-598$.

Lind, S., Xu, R., Stansby, P. K. and Rogers, B. D. (2012a). *Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves.*, J. Comput. Phys. $231(4) : 1499-1523$.

Lind, S., Stansby, P.K. and Rogers, B. D. (2012b). *A multiphase incompressible-compressible smoothed particle hydrodynamics method.*, Proc. 7th Int. SPHERIC Workshop, J.J. Monaghan and J. Kajtar editors. pp $324-332$. May 2012.

Liszka, T. and Orkisz, J. (1980). *The finite difference method at arbitrary irregular grids and its applications in applied Mechanics.*, Computers and Structures, $11 : 83-95$.

Liu, G. R. and Liu, M. B. (2003). *Smoothed Particle Hydrodynamics: a meshfree particle method.*, World Scientific.

Liu, G. R. and Tu Z. H. *MFree2D: an adaptive stress analysis package based on mesh-free technology.*, First MIT Conference and Computational Fluid and Solid Mechanics, $327 - 329$, June 2001, MIT.

Liu, G. R., Liu, M. B. and Lam, K. Y. (2002). *A general approach for constructing smoothing function for meshfree methods.*, Proceedings of the 9th International Conference on Computing in Civil and Building Engineering pp. $431 - 436$, Taipei, Taiwan.

Liu, G. R. and Quek (2003). *The finite element method: a practical course.*, Butterworth Heinemann.

Lobovsky, L., Botia-Vera, E., Castellana, F., Mas-Soler, J., Souto-Iglesias, A. (2014). *Experimental investigation of dynamic pressure loads during dam break*, Journal of Fluids and Structures, Vol: 48, Pagg $407 - 434$.

Lucy, L. B. (1977). *A numerical approach to the testing of the fission hypothesis.*, The Astronomical Journal, vol. 82 n. 12.

Monaghan, J. J. and Lattanzio, J. C. (1985). *A refined particle method for astrophysical problems.*, Astronom. Astrophys. 149, $135 - 143$.

Monaghan, J. J. (1988). *An introduction to SPH.*, Comput. Physics Comun. 48, 89.

Monaghan, J. J. (1989). *On the problem of penetration in particle methods.*, Journal of Computational Physics, 82: 1-15.

Monaghan, J. J. (1990). *Modeling the universe.*, Proc. Astron. Soc. Aust. 18, $233 - 237$.

Monaghan, J. J. (1992). *Smoothed Particle Hydrodynamics.*, Annu. Rev. Astron. Astrophys. 30, $543 - 574$.

Monaghan, J. J. (1994). *Simulating Free Surface Flows with SPH.*, Journal of Computational Physics 110, $399 - 406$.

Panizzo, A. (1994). *Physical and numerical modelling of subaerial landslide generated waves.*, Ph.D. Thesis.

Panizzo, A. and Dalrymple, R. A. (2004). *SPH modelling of underwater landslide generated waves.*, In Proc. 29th International Conference on Coastal Engineering, pages $1147 - 1159$.

Panizzo, A., Cuomo, G. and Dalrymple, R. A. (2006). *3D-SPH simulation of landslide generated waves.*, In Proc. 30th International Conference on Coastal Engineering.

Rogers, B. D. and Dalrymple, R. A. (2008). *SPH modeling of Tsunami Waves*, Advances in Coastal and Ocean Engineering (Series), Edited by: Liu, P.-L.F., Yeh, H. Synolakis, C., Volume: 10, Pages: $75 - 100$, (2008).

Sanz-Serna, J.M. (1992). *Symplectic integrators for Hamiltonian problems: an overview.*, Acta Numerica, vol. 1, $243 - 286$.

Swegle, J. W., Hicks, D. L. and Attaway, S. W. (1995). *Smoothed Particle Hydrodynamics stability analysis.*, Journal of Computational Physics, $116(1) : 123 - 134$.

Plumeraut, L.R. (2009). *Modélisation numérique d'impacts de vagues sur un mur: prise en compte de la presence d'air dans l'eau.*, Thèse Doctorale. Université de Pau et des Pays de l'Adour.

Verlet, L. (1967). *Computer experiments on classical fluids I. Thermodynamical properties of Lennard-Jones molecules.*, Physical Review $159 : 98 - 103$.

Vila, J. P. *On particle weighted methods and smooth particle hydrodynamics.*, Mathematical Models and Methods in Applied Sciences, $09(02) : 161 - 209$, Mar. 1999

Vladimirov, V. S. (1981). *Le distribuzioni nella Fisica Matematica.*, MIR.

Wilkins, M. L. (1999). *Computer simulation of dynamic phenomena.*, Springer-Verlag, Berlin Heidelberg.

Zienkiewicz, O. C. and Taylor, R. L. (2000). *The finite element method.*, 5th ed. Butterworth Heinemann.