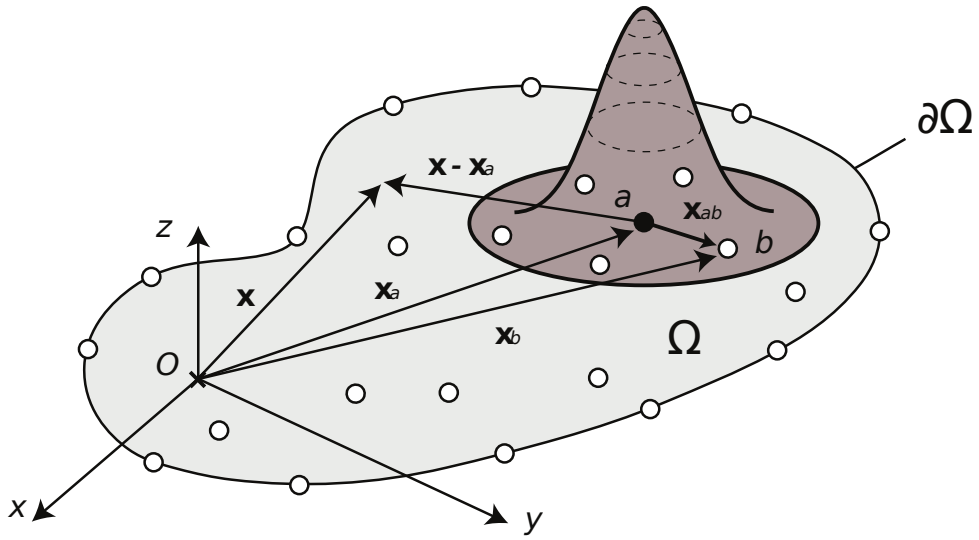
Particle Based Methods Mini-Project
SPH treatment of boundaries and application to moving objects

Marco Sutti
Master in Computational Science and Engineering
École Polytechnique Fédérale de Lausanne
Lausanne, SUISSE
marco.sutti@epfl.ch



May 22, 2019

Contents

List of Figures III

1 SPH approximation of Navier-Stokes equation 3
- 1.1 SPH approximation of density . . . 3
  - 1.1.1 Summation density . . . 3
  - 1.1.2 Continuity density . . . 3
- 1.2 SPH approximation of the equations of motion . . . 4
- 1.3 SPH approximation of $\mathbf{D}'$ . . . 4
- 1.4 SPH approximation of energy . . . 5

2 Boundary treatment 7
- 2.1 Ghost particles . . . 8
  - 2.1.1 Pros and contras . . . 10
- 2.2 Lennard-Jones potential . . . 10
  - 2.2.1 Implementation . . . 11
  - 2.2.2 Simulations . . . 12
  - 2.2.3 Pros and contras . . . 14
- 2.3 Boundary force approach . . . 15
  - 2.3.1 Normal force . . . 15
  - 2.3.2 Simulations . . . 20
  - 2.3.3 Pros and contras . . . 20
  - 2.3.4 Radial force . . . 23

3 Floating objects 25
- 3.1 Equations of motion . . . 25

Appendices 31

A Smoothed particle hydrodynamics 33
- A.1 Integral representation of a function . . . 33
- A.2 SPH approximation of the value of a function . . . 33

B Navier-Stokes equations 35
- B.1 Continuity equation . . . 35
- B.2 Constitutive equations for fluids . . . 35
  - B.2.1 Navier-Poisson law of compressible viscous fluids . . . 36
  - B.2.2 Compressible viscous fluids with no bulk viscosity . . . 36
  - B.2.3 Incompressible viscous fluids . . . 37
- B.3 Equations of motion . . . 37
  - B.3.1 Compressible fluids with bulk viscosity . . . 37
  - B.3.2 Compressible fluids with no bulk viscosity . . . 38
  - B.3.3 Incompressible flows . . . 38

B.4 Incompressible vs compressible flow analysis 38
B.5 Energy equation 39
B.6 Equation of state 39
B.6.1 Equation of state for gas 40
B.6.2 Equation of state for artificial water 40

Bibliography **43**

List of Figures

2.1 Behaviour of SPH near the boundaries. 7
2.2 Stability of two-dimensional SPH. 8
2.3 Construction of VP of type II according to (Liu and Liu, 2003). 9
2.4 Types of particles falling into the support domain of a particle a close to the boundary. 9
2.5 A graph of strength versus distance for the 12-6 Lennard-Jones potential. . 10
2.6 Case 1 of simulation with Lennard-Jones repulsive forces. 12
2.7 Case 2 of simulation with Lennard-Jones repulsive forces. 13
2.8 Case 3 of simulation with Lennard-Jones repulsive forces. 13
2.9 Normal boundary force approach 15
2.10 Case 1 of simulation with normal boundary forces (Monaghan, Kos and Issa, 2003). 21
2.11 Case 2 of simulation with normal boundary forces (Monaghan and Kos, 1999). 21
2.12 Ambiguity in the calculation of the outward normal at a corner. 22
2.13 Radial boundary force approach 23

3.1 Floating object in a two-dimensional flow. 25
3.2 Calculation of the force on each boundary particle of the rigid body. 26
3.3 Calculation of the boundary force: force normal to the boundary. 26

A.1 Smoothed particle hydrodynamics. 34

Introduction and outline

Since the birth of SPH, several boundary treatments have been proposed over the years. In class we did not have time to go deeper into this interesting and delicate aspect of SPH. That is why this mini-project aims at presenting the most established approaches for boundary treatment along with some recent developments.

The report begins with an essential overview of the SPH approximation of Navier-Stokes equations.

Subsequently, the features of the most common SPH boundary treatments are pointed out and some implementations in an SPH code are highlighted.

Since boundaries are also particularly interesting when they are moving, the equations of motion for floating objects are presented along with their SPH approximation.

This mini-project mainly follows a theoretical perspective, but there has also been room for some simple implementations of SPH boundary treatment approaches and relative simulations. These numerical experiments have made use of a MATLAB version of the SPH code that can be found in the book by Liu and Liu (2003), which I implemented during previous research studies.

In summary, the mini-project will basically comprise the following chapters:

1. Overview of SPH discretization of Navier-Stokes equations;
2. SPH treatments of boundaries;
3. Applications to floating bodies.

There are also two appendices, which include:

- An overview of the SPH discretization technique;
- An overview of the Navier-Stokes equations.

The goal of this mini-project is to carry out a critical review of some of the boundary treatment approaches that can be found in the literature and to understand something more about the mathematics which is behind the SPH method.

Please also note that all the figures in this mini-project are original and have been realised using Adobe Illustrator and Inkscape.

Chapter 1

SPH approximation of Navier-Stokes equation

In this chapter we will see how to derive the SPH approximation of the Navier-Stokes equations. This is not a mere mathematical exercise, since the introduction of the boundary treatment that we will see in the next chapter is strictly related to the equations of motion. However, I tried to keep this chapter rather short in order to quickly reach chapter two, which actually represents the core of this project.

For the sake of clarity (and brevity), the Navier-Stokes equations have been reported in Appendix B for the reader who wants to make a comparison between their classical continuous form and the corresponding SPH-approximation.

1.1 SPH approximation of density

The SPH approximation of density can be performed with two different approaches:

- summation density;
- continuity density.

1.1.1 Summation density

The SPH approximation of density of particle a with the summation density approach is:

$$\rho_a = \sum_{b=1}^{N} m_b W_{ab}$$

here W_{ab} is a shorter notation for $W(|\mathbf{x_a} - \mathbf{x_b}|, h)$ and N is the number of particles that fall into the support domain of particle a. This formulation is preferred for general fluid phenomena and it has the advantage of conserving the mass exactly, but it shows edge effects for particles at the interface between two materials and at the boundaries (boundary particle deficiency).

1.1.2 Continuity density

The continuity density approach is the SPH approximation of the continuity equation (B.1):

$$\boxed{\frac{\mathrm{D}\rho_a}{\mathrm{D}t} = -\rho_a \sum_{b=1}^{N} \frac{m_b}{\rho_b} v_j^{ba} \frac{\partial W_{ab}}{\partial x_j^a}}$$

with $j = 1, 2, 3$, in three dimensions. The introduction of the velocity difference v_j^{ba} takes into account for the relative velocities between two particles, so that the density only changes when particles are in relative motion. This approach to the approximation of density is usually preferred since it is also able to deal with discontinuous phenomena, such as explosions or wave-breaking.

1.2 SPH approximation of the equations of motion

Using the SPH approach, we will discretize the equations of motion for compressible fluids with no bulk viscosity (see section B.3.2), namely equation (B.11):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \mathbf{T}' + \rho \mathbf{b}$$

If we neglect the body forces, we multiply both sides by ρ and we substitute $\mathbf{T}' = 2\mu\mathbf{D}'$, we get:

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\frac{\boldsymbol{\nabla} p}{\rho} + \frac{1}{\rho}\boldsymbol{\nabla} \cdot (2\mu \mathbf{D}')$$

In indicial notation:

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = -\frac{1}{\rho} p_{,i} + \frac{2}{\rho}\mu\, D'_{ij,j}$$

with the indices $i, j = 1, 2, 3$. We dropped the summation sign since we are using Einstein's notation.

The SPH approximation of this equation is:

$$\boxed{\frac{\mathrm{d}v_i^a}{\mathrm{d}t} = -\sum_{b=1}^{N} m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_i^a} + 2\sum_{b=1}^{N} m_b \left(\frac{\mu_a D_{ij}'^a}{\rho_a^2} + \frac{\mu_b D_{ij}'^b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_j^a}} \tag{1.1}$$

or, if we use the second form of symmetrization:

$$\boxed{\frac{\mathrm{d}v_i^a}{\mathrm{d}t} = -\sum_{b=1}^{N} m_b \left(\frac{p_a + p_b}{\rho_a \rho_b}\right) \frac{\partial W_{ab}}{\partial x_i^a} + 2\sum_{b=1}^{N} m_b \frac{\mu_a D_{ij}'^a + \mu_b D_{ij}'^b}{\rho_a \rho_b} \frac{\partial W_{ab}}{\partial x_j^a}}$$

Either we choose the first or the second form of symmetrization, we always have to deal with three scalar equations (if we are working in three-dimensional space).

In the above equations, v_i^a denotes the particle a velocity component in the i-direction. $D_{ij}'^a$ is the deviator of the rate of deformation tensor for particle a. Note the repeated index j in the second term on the RHS of the previous equations: this stands for summation over $j = 1, 2, 3$, according to Einstein's notation.

1.3 SPH approximation of $\mathbf{D}'$

To complete the SPH approximation of the equations of motion, we need to know how to compute the deviator of the rate of deformation tensor, $D_{ij}'^a$. The rate of deformation tensor is defined as (equation (B.3)):

$$\mathbf{D}' = \mathbf{D} - \frac{1}{3}\mathrm{tr}(\mathbf{D})\mathbf{I}$$

Moreover, $\mathbf{D} = \dfrac{\boldsymbol{\nabla}\mathbf{v} + \mathbf{v}\boldsymbol{\nabla}}{2}$ and $\mathrm{tr}(\mathbf{D}) = \boldsymbol{\nabla} \cdot \mathbf{v}$, therefore:

$$\mathbf{D}' = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{v} + \mathbf{v}\boldsymbol{\nabla}) - \frac{1}{3}(\boldsymbol{\nabla} \cdot \mathbf{v})\mathbf{I}$$

In indicial notation, this expression becomes:

$$D'_{ij} = \frac{1}{2}\left(v_{i,j} + v_{j,i}\right) - \frac{1}{3}\left(v_{k,k}\right)\delta_{ij}$$

As a side remark, note that in three dimensions $\mathbf{D'}$ has 9 components, but 6 of them are equal to each other because of symmetry properties, so in general one can consider just 6 components. Analogously, in two dimensions, $\mathbf{D'}$ has 4 components, but 2 of them are equal to each other because of symmetry properties, so one can consider just 3 components.

The SPH approximation for D'^a_{ij} is thus:

$$\boxed{D'^a_{ij} = \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_j^{ba}\,\frac{\partial W_{ab}}{\partial x_i^a} + \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_i^{ba}\,\frac{\partial W_{ab}}{\partial x_j^a} - \left(\frac{1}{3}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_k^{ba}\,\frac{\partial W_{ab}}{\partial x_k^a}\right)\delta_{ij}} \tag{1.2}$$

Note that in the last term of this expression a summation over the k index is implied because of the usual convention for repeated indices.

We could expand eq. (1.2) and get all the six components of $\mathbf{D'^a}$:

$$D'^a_{11} = \sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} - \frac{1}{3}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\left(v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} + v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a} + v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a}\right)$$

$$D'^a_{22} = \sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a} - \frac{1}{3}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\left(v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} + v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a} + v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a}\right)$$

$$D'^a_{33} = \sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a} - \frac{1}{3}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\left(v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} + v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a} + v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a}\right)$$

$$D'^a_{12} = \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} + \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a}$$

$$D'^a_{13} = \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_1^a} + \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_1^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a}$$

$$D'^a_{23} = \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_3^{ba}\,\frac{\partial W_{ab}}{\partial x_2^a} + \frac{1}{2}\sum_{b=1}^{N}\frac{m_b}{\rho_b}\,v_2^{ba}\,\frac{\partial W_{ab}}{\partial x_3^a}$$

where we have used the fact that the velocity difference vector $\mathbf{v^{ba}} = \mathbf{v^b} - \mathbf{v^a}$ has the components $\mathbf{v^{ba}} \equiv (v_1^{ba}, v_2^{ba}, v_3^{ba}) \equiv (v_1^b - v_1^a, v_2^b - v_2^a, v_3^b - v_3^a)$ and the position vector $\mathbf{x^a}$ of particle a has the components $\mathbf{x^a} \equiv (x_1^a, x_2^a, x_3^a)$.

1.4 SPH approximation of energy

The equation of energy for compressible viscous fluids with no bulk viscosity is (equation (B.15)):

$$\rho\,\frac{\mathrm{d}e}{\mathrm{d}t} = -p\boldsymbol{\nabla}\cdot\mathbf{v} + 2\mu\mathbf{D'} : \boldsymbol{\nabla}\mathbf{v}$$

in indicial notation:

$$e_{,t} = \frac{1}{\rho}\left(-p\,v_{k,k} + 2\mu D'_{ij}v_{i,j}\right)$$

where $i, j, k = 1, 2, 3$.

The SPH approximation of the energy equation for particle a is:

$$\boxed{\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{2}\sum_b^N m_b \left(\frac{p_a + p_b}{\rho_a \rho_b}\right) v_j^{ab} \frac{\partial W_{ab}}{\partial x_j^a} + 2\,\frac{\mu_a}{\rho_a}\left(D_{ij}'^a\right)^2}$$

or, with the second form of symmetrization:

$$\boxed{\frac{\mathrm{d}e_a}{\mathrm{d}t} = \frac{1}{2}\sum_b^N m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) v_j^{ab} \frac{\partial W_{ab}}{\partial x_j^a} + 2\,\frac{\mu_a}{\rho_a}\left(D_{ij}'^a\right)^2}$$

We can write the first form in extenso to inspect its structure:

$$\frac{\mathrm{d}e_a}{\mathrm{d}t} = \underbrace{\frac{1}{2}\sum_b^N m_b \left(\frac{p_a + p_b}{\rho_a \rho_b}\right) \underbrace{\left[v_1^{ab}\frac{\partial W_{ab}}{\partial x_1^a} + v_2^{ab}\frac{\partial W_{ab}}{\partial x_2^a} + v_3^{ab}\frac{\partial W_{ab}}{\partial x_3^a}\right]}_{\text{divergence of the velocity difference}}}_{\text{pressure work}}$$
$$+ \underbrace{2\,\frac{\mu_a}{\rho_a}\left[\left(D_{11}'^a\right)^2 + \left(D_{22}'^a\right)^2 + \left(D_{33}'^a\right)^2 + \underbrace{2\left(D_{12}'^a\right)^2 + 2\left(D_{23}'^a\right)^2 + 2\left(D_{13}'^a\right)^2}_{\text{because of symmetry of } \mathbf{D}'}\right]}_{\text{energy dissipation due to viscous forces (viscous entropy)}}$$

This completes the presentation of the SPH approximation of Navier-Stokes equations.

Chapter 2

Boundary treatment

SPH has been unable to treat generalised boundary conditions, but this is not an inherent limitation of the method.

— Randles and Libersky, 1996

In the early CFD uses of SPH, simple boundary conditions such as non-penetrating surfaces were adopted. Nonetheless, being a particle method, the boundary of the simulation domain is never well defined. The problem is that close to the boundaries of the simulation domain, the SPH method is affected by particle deficiency, since the integral of the kernel function is truncated by the boundary (figure 2.1).

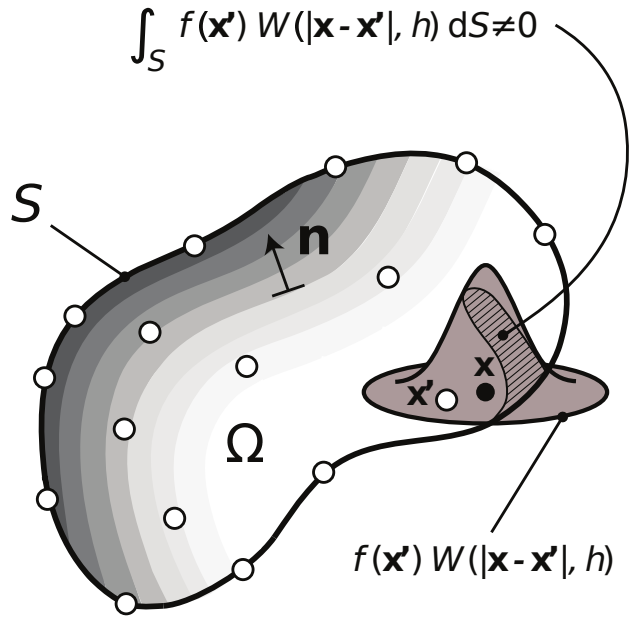


Figure 2.1: Behaviour of SPH near the boundaries.

In fact, the SPH method needs a sufficient and necessary number of particles within the support domain of κh in order to be stable. In one, two and three dimensions the number of neighbouring particles (including the particle itself) should be about 5, 21 and 57, respectively, if the particles are arranged in an initial lattice with a smoothing length of 1.2 times the particle spacing, and $\kappa = 2$ (Liu and Liu, 2003). Figure 2.2 illustrates this concept for the two-dimensional case.

Therefore, it is evident that for particles near or on the boundary only particles inside the boundary contribute to the summation, and this one-sided contribution leads to wrong

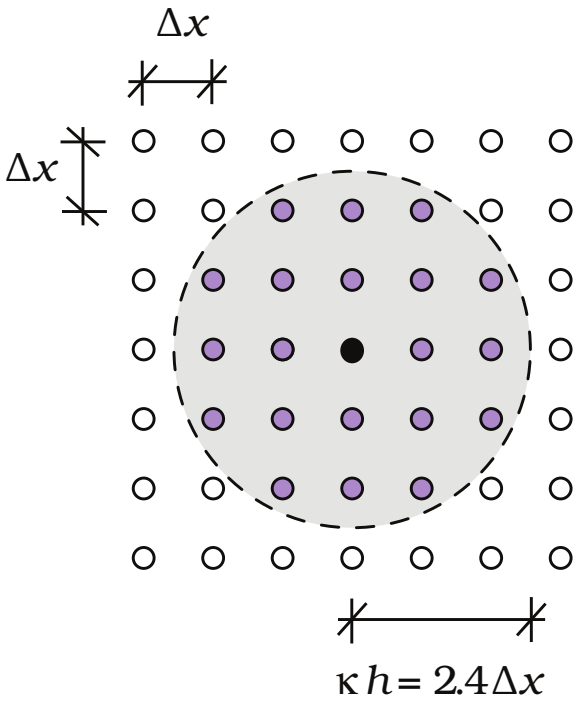


Figure 2.2: Stability of two-dimensional SPH.

solutions.

In this chapter we will see how over the years several authors have proposed different solutions to address the problem of boundary treatment in SPH. Basically, in all of these approaches, boundaries are usually defined by lines of particles that exert repulsive forces on fluid particles. In particular, in what follows we will explore the following concepts and formulations:

- Ghost particles (Libersky, 1993);
- Lennard-Jones potential for repulsive boundary forces (Monaghan, 1994);
- Normal boundary forces (Monaghan and Kos, 1999; Monaghan, Kos and Issa, 2003);
- Radial boundary forces (Monaghan and Kajtar, 2009).

2.1 Ghost particles

One of the first solutions some authors came up with is the introduction of *ghost particles*.

In 1993, Libersky was the first to introduce ghost particles to reflect a symmetrical surface and in 1994 Monaghan used a line of virtual particles located right on the solid boundary, to produce a *highly repulsive force* to the particles near the boundary, in order to avoid penetration.

Liu et al. (2001-2002) merged and improved these two approaches to treat the solid boundary conditions (Liu and Liu, 2003). According to them, virtual particles (VP) can be of two types:

- Type I: right on the solid boundary (similar to Monaghan, 1994);
- Type II: they fill in the region outside the boundary and close to it (similar to what Libersky used in 1993).

VP of type I take part in kernel and particle approximation for the real particles. Their position and physical variables do not evolve in the simulation and they are used to exert a **repulsive boundary force** to prevent the interior particles from penetrating the solid boundary.

The ghost particles (i.e., VP of type II) can be applied to both *solid boundary* and *free surfaces*. They are constructed as follows: if a real particle i is located within the distance κh from the boundary, then a VP is symmetrically placed on6 the outside of the boundary. These particles have the *same density* and *pressure* as the corresponding real particles, but *opposite velocity* (see figure 2.3). VP of type II do not evolve their parameters, since they are created symmetrically to the corresponding real particles at every evolution step.

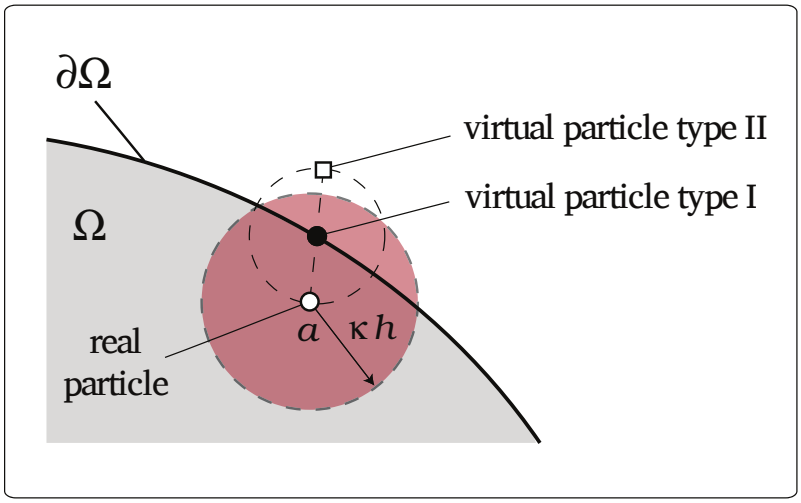


Figure 2.3: Construction of VP of type II according to (Liu and Liu, 2003).

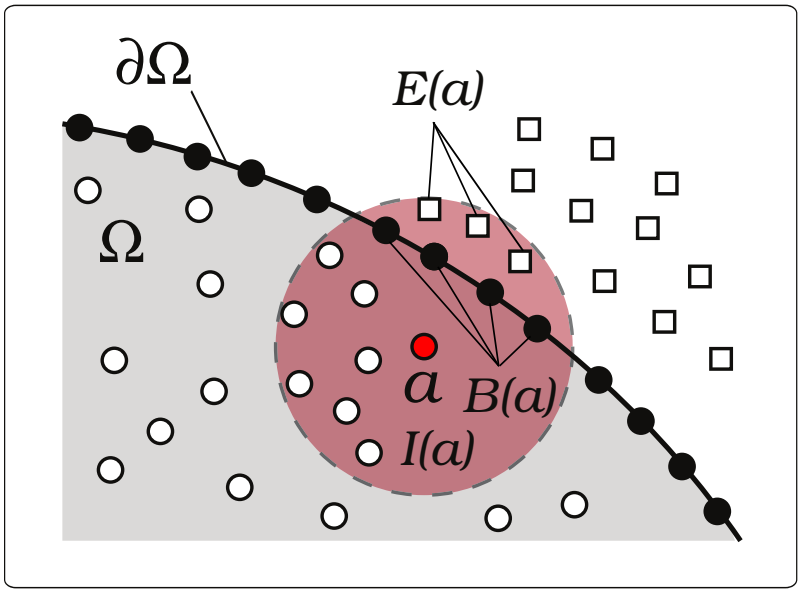


Figure 2.4: Types of particles falling into the support domain of a particle a close to the boundary.

Summarizing, for a particle a close to the boundary, the set of particles falling into its domain of influence can be divided into three subsets (see figure 2.4):

- $I(a)$: interior or real particles;

- $B(a)$: boundary particles, or virtual particles of type I;
- $E(a)$: exterior particles, or virtual particles of type II (ghost particles).

The total number of particles N in the support domain of a particle a close to the boundary is therefore given by $N(a) = I(a) \cup B(a) \cup E(a)$.

2.1.1 Pros and contras

This approach has the advantages to restore the SPH consistency near the boundaries and to prevent non-physical penetration through the solid boundary. Yet in the original presentation of Randles and Libersky (1996), a very important point was missing: *how to generate* the ghost particles, and still nowadays it is not clear how to address this issue. In the most usual approach, the ghost particles are generated by mirroring the fluid particles, and this implies that they must adapt with the fluid particles at each time-step, causing an additional computational effort. The ghost particles become particularly unwieldy in presence of corners or surfaces with high curvature, since in such situations they cannot be placed without ambiguity.

2.2 Lennard-Jones potential

The formulation of this repulsive force has been guided by the known forces between molecules (Monaghan, 1994): in fact, it takes the form of a **Lennard-Jones potential**. An example of Lennard-Jones potential is shown in figure 2.5.

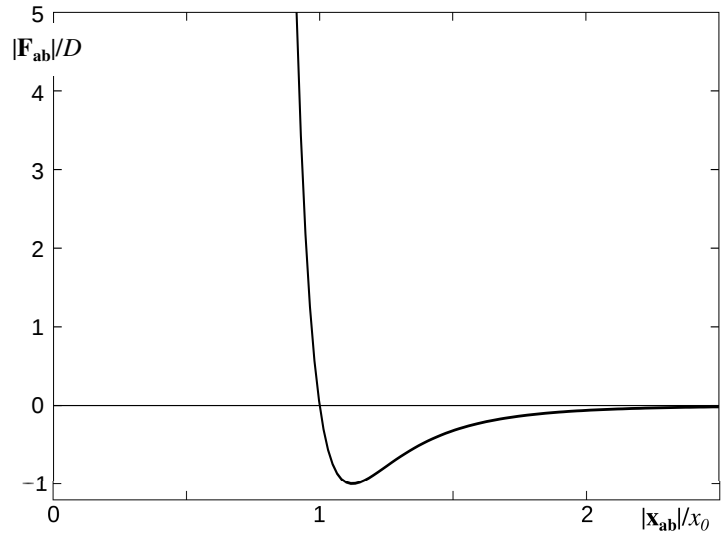


Figure 2.5: A graph of strength versus distance for the 12-6 Lennard-Jones potential.

The way this potential works is quite simple: if a real particle is approaching a VP of type I (i.e. a boundary particle), then a pairwise repulsive force is applied along the centreline of these two particles:

$$\mathbf{F_{ab}} = \begin{cases} D\left[\left(\dfrac{x_0}{|\mathbf{x_{ab}}|}\right)^{n_1} - \left(\dfrac{x_0}{|\mathbf{x_{ab}}|}\right)^{n_2}\right] \dfrac{\mathbf{x_{ab}}}{|\mathbf{x_{ab}}|^2} & |\mathbf{x_{ab}}| \leq x_0 \\ 0 & |\mathbf{x_{ab}}| > x_0 \end{cases}$$

where:

- $|\mathbf{x_{ab}}|$ is the distance between particle a and particle b;

- x_0 is a cut-off distance;
- D is a problem parameter;
- usually $n_1 = 12$ and $n_2 = 6$, but also $n_1 = 4$ and $n_2 = 2$.

The coefficient D should be chosen considering the physical configuration. For problems involving dams, bores, weirs with fluid depth H, we may set $D = 5gH$, but also $D = 10gH$ or $D = gH$.

The cut-off distance x_0 is usually selected to be approximately equal to the initial particle spacing Δx. If it is too large, then some particles may feel the repulsive force in the initial distribution already. If it is too small, then the particles will penetrate the boundary before feeling the repulsive force. In the simulations that I ran, a cut-off distance equal to $x_0 = 0.48\,\Delta x$ seemed to provide good results.

2.2.1 Implementation

Small implementations have also been an integral part of this project, hence the piece of code reported hereafter is here not just to "fill in the space", but as a proof of the work that has been done.

The code that I have modified is a MATLAB version of the one discussed in the book by Liu and Liu (2003), and that I implemented during my previous research studies.

```
if boundary_force_approach==1
%=========================================================================
% Opt. 1: Boundary particle force and penalty anti-penetration force
% [Monaghan, 1994] in Lennard-Jones form.
%=========================================================================

r_0 = coeff_r_0*part_spac;

for k=1:n_interact_pairs   %For every interacting pair...
    i = pair_i(k);   %take the index of the 1st particle in the pair
    j = pair_j(k);   %take the index of the 2nd particle in the pair
    if(part_type(i)*part_type(j)<0)  %if i & j are of different type
        r_ij = 0;               %compute the distance between i and j
        for d=1:dim
            x_ij(d) = x(i,d) - x(j,d);
               %x_ij(d): distance between i and j in the d-direction
            r_ij = r_ij + x_ij(d)*x_ij(d);
        end
        r_ij = sqrt(r_ij);             %distance between i and j

        if(r_ij<r_0)
        %If the distance between i and j is less than the cutoff...
        %... compute the repulsive force per unit mass
            f = D*((r_0/r_ij)^n_1-(r_0/r_ij)^n_2)/r_ij^2;

     %Add the repulsive force to the acceleration in the d-direction
            for d = 1:dim
                ext_dvdt(i,d) = ext_dvdt(i,d) + f*x_ij(d);
            end

        end

    end

end
```

2.2.2 Simulations

In the simulations that I ran, I considered the classical dam-break problem, where a column of liquid is going to fall under the effect of gravity, as if a barrier containing the liquid were suddenly removed, like in the breaking of a dam. The column of fluid has a height of 40 metres and a width of 20 metres, and it is discretized with 600 SPH particles. The Lennard-Jones potential has $n_1 = 4$ and $n_2 = 2$.

In what follows, we show some snapshots from the simulations performed for different values of the cut-off distance. Obviously, our attention is mainly focused on the behaviour of the particles at the boundary.

Case 1. In the first case we considered a very large cut-off distance, equal to twice the initial particle spacing. Figure 2.6 reports a snapshot from this simulation and it clearly that such a cut-off distance not only causes large repulsive forces at the boundaries, but also brings about numerical instabilities propagating throughout the simulation domain.

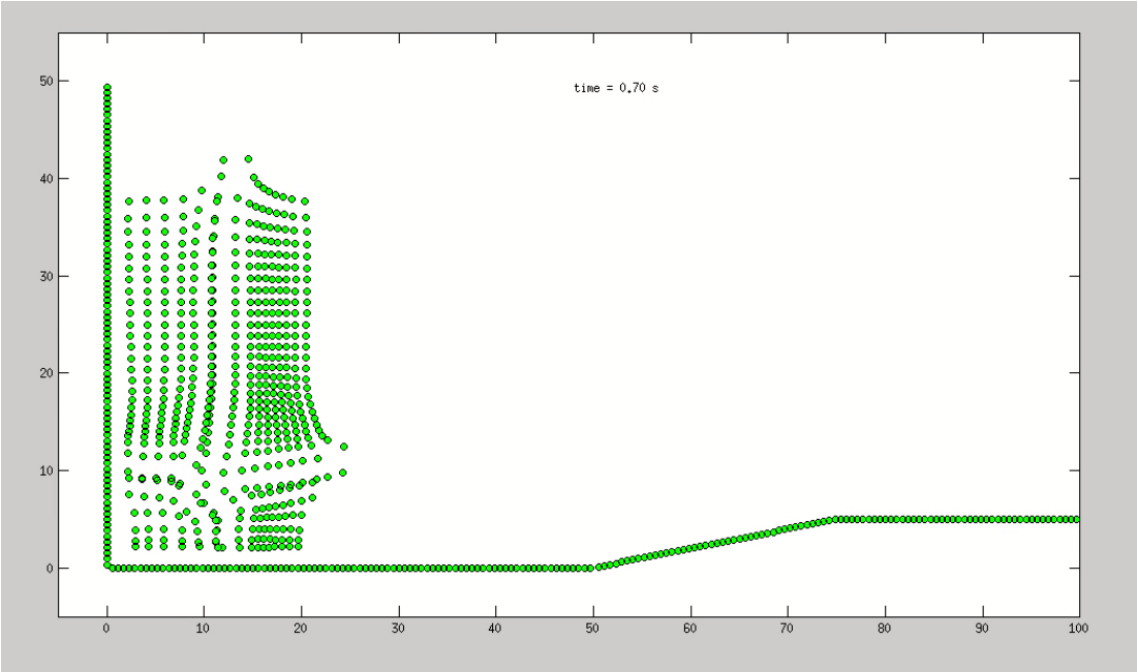


Figure 2.6: Case 1 of simulation with Lennard-Jones repulsive forces.

Case 2. In this case we considered a very small cut-off distance, equal to $0.05\,\Delta x$. Figure 2.7 reports a snapshot from this simulation which clearly shows nasty and non-physical behaviour of fluid particles penetrating the solid boundaries.

Case 3. In this case we considered a cut-off distance equal to $0.48\,\Delta x$, which is a value that seemed to provide good results. Anyway, it can be easily seen from the simulation snapshot (at $t = 1.98$ s) reported in figure 2.8 that there are large, non-physical boundary layers, both along the vertical and the horizontal boundaries. This is an intrinsic flaw of the Lennard-Jones formulation of the repulsive boundary forces.

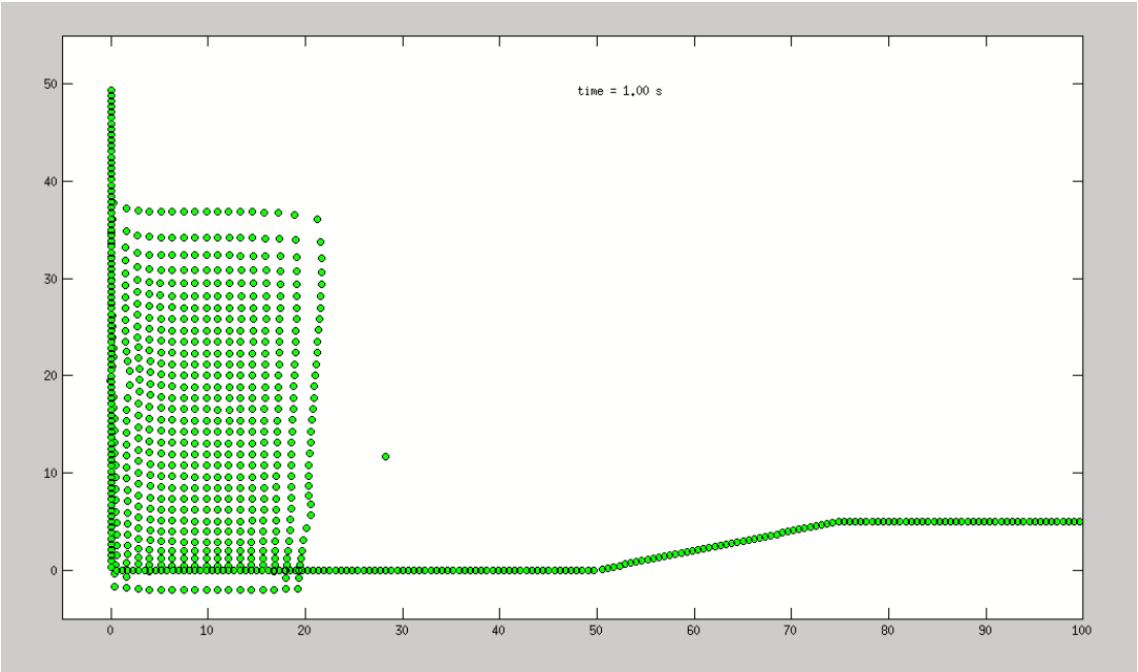


Figure 2.7: Case 2 of simulation with Lennard-Jones repulsive forces.

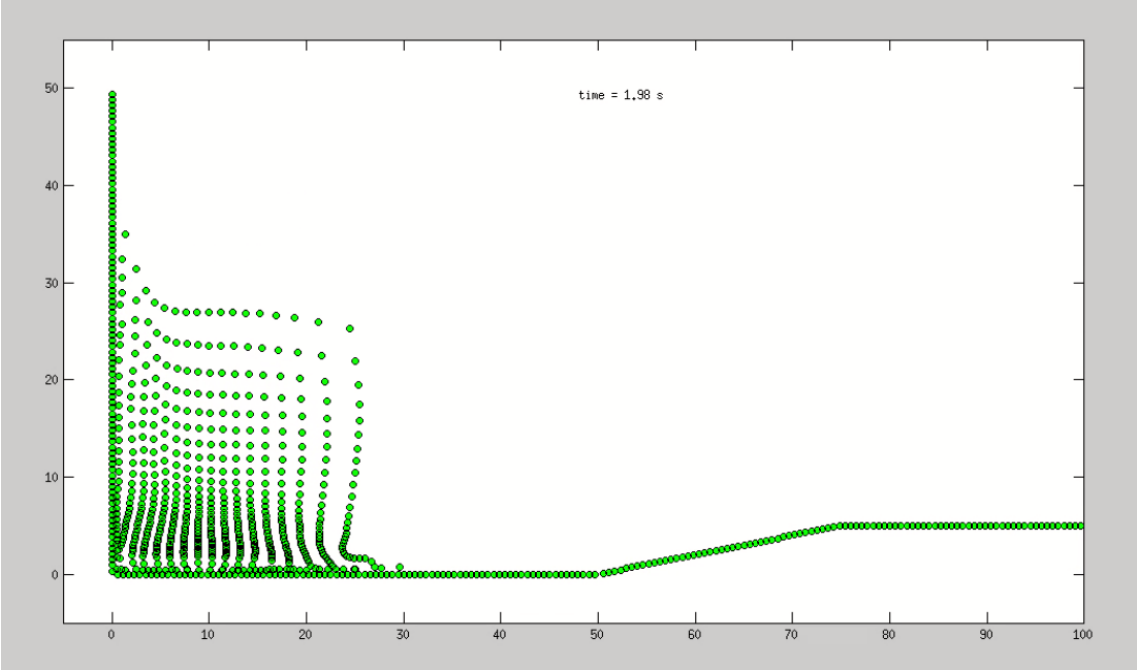


Figure 2.8: Case 3 of simulation with Lennard-Jones repulsive forces.

2.2.3 Pros and contras

As we have seen, if the parameters are well tuned, the Lennard-Jones potential formulation avoids particle penetration into the boundary, ensuring that the velocity component normal to the boundaries vanishes. Nonetheless, Lennard-Jones forces are not satisfactory, since a particle moving parallel to the boundary is subject to a non-uniform normal force and a non-zero tangential force, leading to large disturbances in the flow near a boundary. Because of this defect, some authors adopted boundary particle forces based on an interpolation procedure that we are going to present in the next section.

2.3 Boundary force approach

According to Monaghan et al. (2003), *the force per unit mass* $\mathbf{f_k}$ *on the boundary particles is due to the fluid particles unless the moving rigid body strikes a fixed boundary.* Disregarding the latter case, we can write:

$$\mathbf{f_k} = \sum_a \mathbf{f_{ka}}$$

where $\mathbf{f_{ka}}$ represents the force per unit mass on boundary particle k due to fluid particles a. The force $\mathbf{f_k}$ on each boundary particle is computed by summing up the contributions from all the surrounding water particles which fall within the supporting kernel (see figure 3.2).

In the following, we will review two different formulations available in literature for the calculation of the boundary force $\mathbf{f_{ka}}$.

2.3.1 Normal force

Monaghan et al. (2003) proposed that the forces from neighbouring boundary particles should give rise to a force normal to the boundary. Let's consider the k-th boundary particle and the fluid particle a, as shown in figure 2.9.

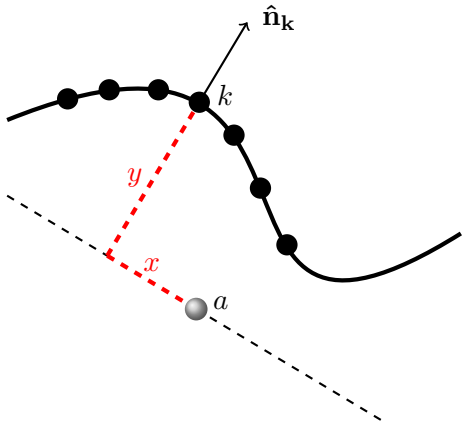


Figure 2.9: Normal boundary force approach

Then the repulsive force per unit mass on boundary particle k due to fluid particle a is given by:

$$\mathbf{f_{ka}} = -\frac{m_a}{m_a + m_k} B(x, y)\, \hat{\mathbf{n}}_\mathbf{k}$$

and, in turn, because of Newton's third law, the force per unit mass on a due to k is:

$$\boxed{\mathbf{f_{ak}} = +\frac{m_k}{m_a + m_k} B(x, y)\, \hat{\mathbf{n}}_\mathbf{k}} \tag{2.1}$$

Let us have a closer look to the parameters involved in these equations:

- $\hat{\mathbf{n}}_\mathbf{k}$ is the outward unit normal to the boundary at the location of particle k;
- m_a and m_k are the masses of particles a and k, respectively;
- x and y are the tangential and the normal distance between a and k, respectively;
- $B(x, y)$ is a function of the local coordinates x and y.

The normal distance y can be viewed as the projection on the direction of $\hat{\mathbf{n}}_\mathbf{k}$ of the vector distance $\mathbf{x_{ak}} = \mathbf{x_k} - \mathbf{x_a}$ between a and k, thus it can be obtained from the following dot product:

$$y = \mathbf{x_{ak}} \cdot \hat{\mathbf{n}}_\mathbf{k}$$

Finally the total force per unit mass on fluid particle a due to all the boundary particles k in the support domain of a will be:

$$\boxed{\mathbf{f_a} = \sum_{k \in B(a)} \mathbf{f_{ak}}} \tag{2.2}$$

where $B(a)$ designates the set of boundary particles falling into the support domain of particle a (see section 2.2 and figure 2.4). This force is then simply added to the SPH approximation of the equations of motion (eq. (1.1)):

$$\frac{\mathrm{d}v_i^a}{\mathrm{d}t} = -\sum_{b=1}^{N} m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_i^a} + 2\sum_{b=1}^{N} m_b \left(\frac{\mu_a D_{ij}'^a}{\rho_a^2} + \frac{\mu_b D_{ij}'^b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_j^a} + f_i^a \tag{2.3}$$

with $i = 1, 2, 3$ if we are working in three dimensions.

Choice of the function $B(x, y)$

We still have to define the form of the function $B(x, y)$ appearing in (2.1). In general, the variation of $B(x, y)$ with x should ensure that the force on a fluid particle moving parallel to the boundary is constant. Moreover, $B(x, y)$ should be chosen so that it rapidly increases as y tends to zero, i.e. as the fluid particle is approaching the boundary, to prevent penetration of the boundaries.

Choice 1. Monaghan et al. (2003) wrote $B(x, y)$ as a separable variable function:

$$B(x, y) = \Gamma(y)\, \chi(x)$$

where

$$\chi(x) = \begin{cases} \left(1 - \dfrac{x}{\Delta p}\right) & \text{if} \quad 0 < x < \Delta p \\ 0 \quad \text{otherwise} \end{cases}$$

with Δp being the spacing between the boundary particles[1]. Note that the function $\chi(x)$ ensures that a fluid particle moving parallel to the boundary will experience a constant force. The function $\Gamma(y)$ has a form related to the gradient of the kernel (Monaghan et al., 2003; Monaghan, 2005):

$$\Gamma(y) = \begin{cases} \frac{2}{3}\,\beta & \text{if} \quad 0 < q < \frac{2}{3} \\ \beta\left(2q - \frac{3}{2}\,q^2\right) & \text{if} \quad \frac{2}{3} < q < 1 \\ \frac{1}{2}\,\beta\,(2 - q)^2 & \text{if} \quad 1 < q < 2 \\ 0 \quad \text{otherwise} \end{cases}$$

where:

- $q = y/h$, being h the smoothing length;

[1] Usually the boundary particle spacing Δp is about one half of the initial fluid particle spacing Δx (Monaghan and Kajtar, 2009).

- $\beta = 0.02\, c_s^2/y$, being c_s the speed of sound. This term represents an estimate of the maximum force per unit mass necessary to stop a particle moving at the estimated maximum speed. The factor $1/y$ ensures that a faster moving particle can be stopped.

Overall, the function $\Gamma(y)$ quickly increases as y decreases in order to prevent the fluid particle from penetrating the boundary of the simulation domain.

The following piece of code represents the implementation of the normal boundary force approach using the formulation of Monaghan et al. (2003).

```
elseif boundary_force_approach==2
%=========================================================================
% Opt. 2: Normal Boundary Repulsive force according to Monaghan,
% Kos and Issa (2003)
%=========================================================================

for k=1:n_interact_pairs   %For every interacting pair...
    i = pair_i(k);   %take the index of the 1st particle in the pair
    j = pair_j(k);   %take the index of the 2nd particle in the pair
    if(part_type(i)*part_type(j)<0)  %if i & j are of different type
        r_ij = 0;               %compute the distance between i and j
        for d=1:dim
            x_ij(d) = x(i,d) - x(j,d);
               %x_ij(d): distance between i and j in the d-direction
            r_ij = r_ij + x_ij(d)*x_ij(d);
        end
        r_ij = sqrt(r_ij);             %distance between i and j

        if part_type(i)<0    % If i is a boundary particle...
            index_bp = i;
            index_fp = j;
        else
            index_bp = j;
            index_fp = i;
        end

        % Mapping the index for all the particles to the index just
        % for the boundary particles:
        index_bp = index_bp - n_real_part;


        % Calculation of the local coordinates x and y
        y_local = abs(x_ij(1)*unit_normal(index_bp,1)
                  + x_ij(2)*unit_normal(index_bp,2));
        x_local = sqrt(r_ij^2-y_local^2);


        % Calculation of the function B(x,y)
        beta = 0.02*c(index_fp)^2/y_local;
        q = y_local/hsml(index_bp);


        % Calculate Gamma(y)
        if (q>=0 && q<=0.66)
            Gamma = 0.66*beta;
        elseif (q>0.66 && q<1)
            Gamma = beta*(2*q-1.5*q^2);
        elseif (q>=1 && q<=2)
```

```
            Gamma = 0.5*beta*(2-q)^2;
        else
            Gamma = 0;
        end

        % Calculate Chi(x)
        if x_local<virt_part_spac
            Chi = 1-x_local/virt_part_spac;
        else
            Chi = 0;
        end

        % Compute B(x,y) = Gamma(y)*Chi(x):
        B = Gamma*Chi;


        f = mass(1)/(mass(1)+mass(end))*B/r_ij;

     %Add the repulsive force to the acceleration in the d-direction
            for d = 1:dim
                ext_dvdt(i,d) = ext_dvdt(i,d) + f*x_ij(d);
            end

        end

    end

end
```

Choice 2. A similar approach was given by Monaghan and Kos (1999), who defined $\mathbf{f_{ak}}$ as:

$$\boxed{\mathbf{f_{ak}} = R(y)\, P(x)\, \hat{\mathbf{n}}_\mathbf{k}}$$

The specific form of $R(y)$ is not crucial, however, as before, it should rapidly increase as y tends to zero. The authors proposed the following formulation:

$$R(y) = \begin{cases} A \dfrac{1}{\sqrt{q}} (1-q) & \text{if} \quad q < 1 \\ 0 & \text{otherwise} \end{cases}$$

where:

- $q = y/2h$, being h the smoothing length;
- A is a coefficient having dimensions of an acceleration:

$$A = \frac{1}{h} 0.01 c_a^2$$

  with c being the speed of sound corresponding to particle a.

As before, the function $P(x)$ is devised so that a fluid particle moving parallel to the boundary will experience a constant repulsive boundary force:

$$P(x) = \begin{cases} \dfrac{1}{2}\left(1 + \cos \dfrac{\pi x}{\Delta p}\right) & \text{if} \quad x < \Delta p \\ 0 & \text{otherwise} \end{cases}$$

where Δp is the boundary particle spacing.

The following piece of code represents the implementation of the normal boundary force approach using the formulation of Monaghan and Kos (1999).

```
elseif boundary_force_approach==3
%=========================================================================
% Opt. 3: Normal Boundary Repulsive force according to Monaghan and
% Kos (1999).
%=========================================================================

for k=1:n_interact_pairs   %For every interacting pair...
    i = pair_i(k);   %take the index of the 1st particle in the pair
    j = pair_j(k);   %take the index of the 2nd particle in the pair
    if(part_type(i)*part_type(j)<0)  %if i & j are of different type
        r_ij = 0;              %compute the distance between i and j
        for d=1:dim
            x_ij(d) = x(i,d) - x(j,d);
               %x_ij(d): distance between i and j in the d-direction
            r_ij = r_ij + x_ij(d)*x_ij(d);
        end
        r_ij = sqrt(r_ij);            %distance between i and j

        if part_type(i)<0    % If i is a boundary particle...
            index_bp = i;
            index_fp = j;
        else
            index_bp = j;
            index_fp = i;
        end

        % Mapping the index for all the particles to the index just
        % for the boundary particles:
        index_bp = index_bp - n_real_part;


        % Calculation of the local coordinates x and y
        y_local = abs(x_ij(1)*unit_normal(index_bp,1)
                  + x_ij(2)*unit_normal(index_bp,2));
        x_local = sqrt(r_ij^2-y_local^2);


        % Calculation of the function B(x,y)
        A = 1/hsml(index_bp)*0.01*c(index_fp)^2;
        q = y_local/(2*hsml(index_bp));


        % Calculate R(y)
        if q<1
            R = A/sqrt(q)*(1-q);
        else
            R = 0;
        end

        % Calculate P(x)
        if x_local<3*virt_part_spac
            P = 0.5*(1+cos(pi*x_local/virt_part_spac));
        else
            P = 0;
```

```
55          end
56
57          % Compute B(x,y) = R(y)*P(x):
58          B = R*P;
59
60
61          f = mass(1)/(mass(1)+mass(end))*B/r_ij;
62
63       %Add the repulsive force to the acceleration in the d-direction
64              for d = 1:dim
65                  ext_dvdt(i,d) = ext_dvdt(i,d) + f*x_ij(d);
66              end
67
68          end
69
70      end
71
72  end
```

Choice 3. In 2010, Gesteira et al. introduced a correction term $\epsilon(z, u_\perp)$ into the original formulation of Monaghan and Kos (1999):

$$\mathbf{f_{ak}} = R(y)\, P(x)\, \epsilon(z, u_\perp)\, \hat{\mathbf{n}}_\mathbf{k}$$

The function $\epsilon(z, u_\perp)$ is used to adjust the magnitude of the repulsive force according to the local water depth and velocity of the fluid particle normal to the boundary. Please note that this is the formulation that has been implemented in the SPHysics code, according to the guide of Gesteira et al. (2010).

2.3.2 Simulations

In this section we show a couple of simulations performed with the normal boundary force approach. The problem set-up is the same as the one we presented in section 2.2.2. We considered the formulations of Monaghan, Kos and Issa (2003) and of Monaghan and Issa (1999), which have been outlined in the previous paragraphs **Choice 1** and **Choice 2**, respectively.

Case 1. In the first case we considered the formulation of Monaghan, Kos and Issa (2003). Figure 2.10 reports a snapshot from this simulation (time-frame at $t = 1.98$ s). It can be noticed that the nasty boundary effects appearing in the same simulation using the Lennard-Jones forces (case 3) have disappeared (compare with figure 2.8).

Case 2. In this case we considered the formulation of Monaghan and Kos (1999). Figure 2.11 (time-frame at $t = 1.98$ s) does not show any appreciable difference with respect to figure 2.10 (time-frame at $t = 1.98$ s), except for a quite bizarre short row of particles moving in front of all the other falling fluid particles.

2.3.3 Pros and contras

The main advantage of using a normal boundary force approach over the Lennard-Jones repulsive forces is that now fluid particles moving parallel to the boundary experience a constant repulsive boundary force. This, as we have seen, is one flaw of the Lennard-Jones repulsive forces which is here avoided, along with other unpleasant boundary effects. Unfortunately, the problem with all the normal boundary force approaches actually lies

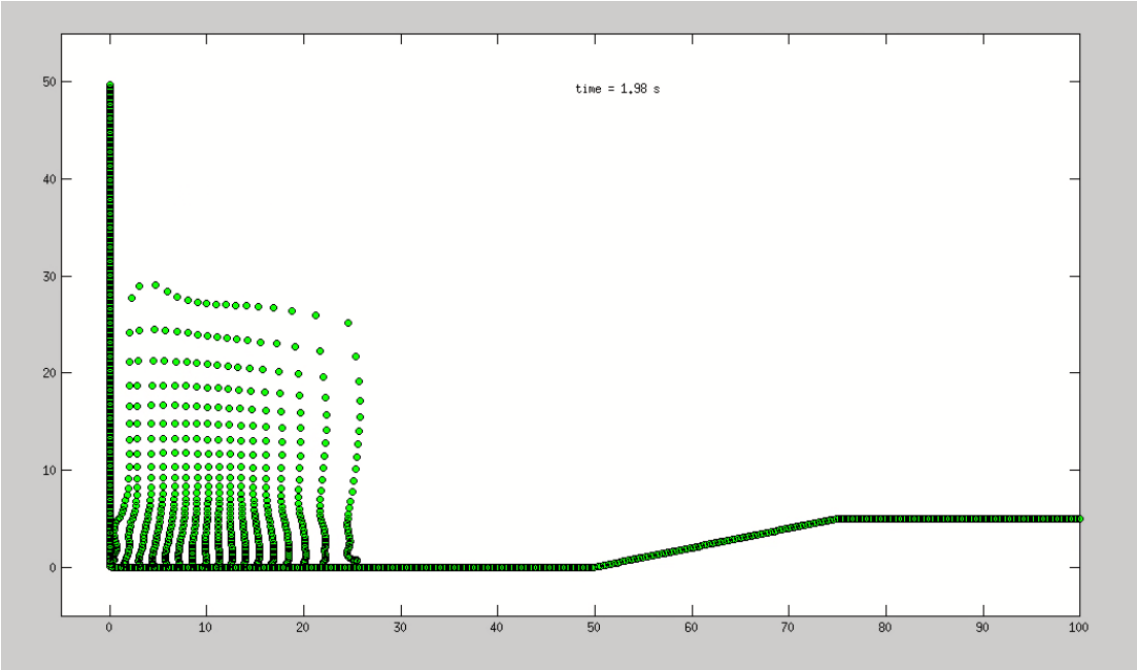


Figure 2.10: Case 1 of simulation with normal boundary forces (Monaghan, Kos and Issa, 2003).

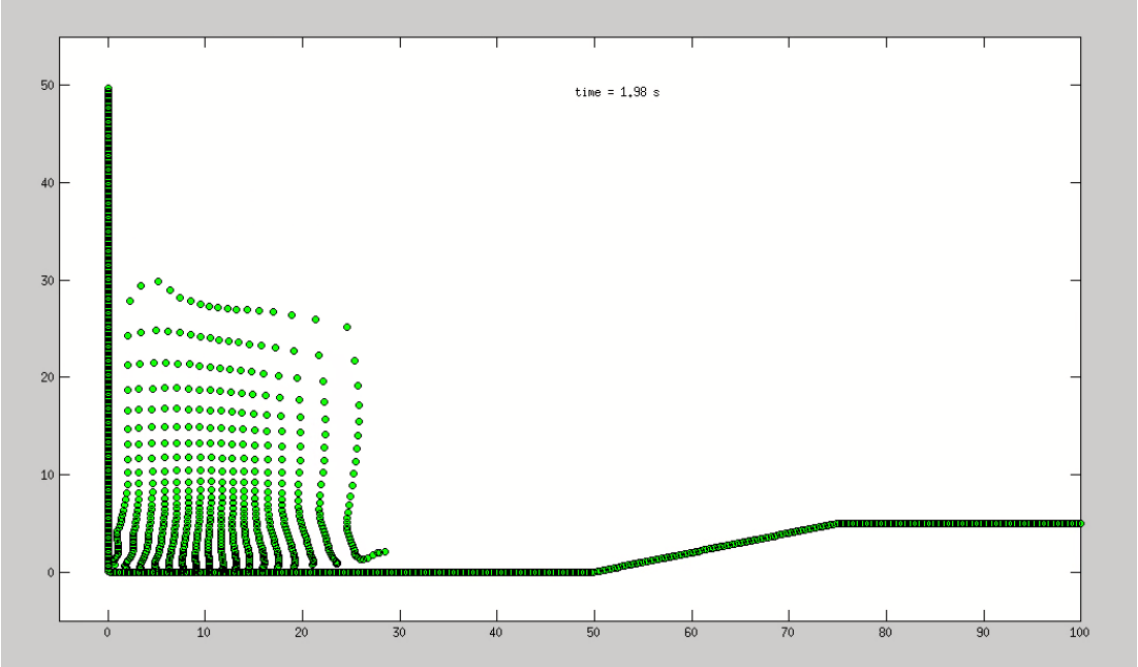


Figure 2.11: Case 2 of simulation with normal boundary forces (Monaghan and Kos, 1999).

in the very calculation of the outward normals to a surface. In fact, this is not such an easy task to be achieved in SPH. Sometimes the calculation of the outward unit normal may be ambiguous, as illustrated in figure 2.12. In this example, a corner particle has two

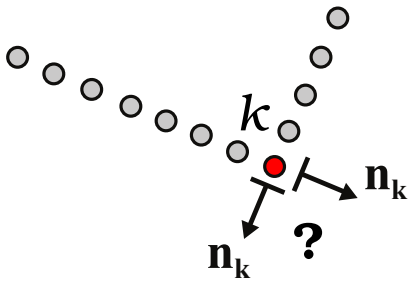


Figure 2.12: Ambiguity in the calculation of the outward normal at a corner.

outward normals, so it may interact with the fluid particles using either one or another of the normals to the lines that make up the corner, or we may assign a normal with direction halfway between the normals to the two straight lines. In any case, it is clear that, because of this ambiguity in defining the outward normal, we have to treat the corner particles differently from the other boundary particles, hence we need to allow the code to identify corner particles. This is not very nice since we will inevitably increase the computational effort of the simulation.

2.3.4 Radial force

In order to solve this problem, ten years later the normal boundary force formulation, Monaghan and Kajtar (2009) suggested to adopt repulsive boundary forces with radial direction (figure 2.13). The proposed formulation for $\mathbf{f_{ak}}$ is:

$$\mathbf{f_{ak}} = \frac{K}{\beta} \frac{\mathbf{x_{ak}}}{|\mathbf{x_{ak}}|^2} W(|\mathbf{x_{ak}}|/h) \frac{2m_k}{m_a + m_k} \tag{2.4}$$

where:

- we choose $K = gD$, with g being the gravitational acceleration and D the initial water depth;
- β is the ratio between the initial fluid particle spacing Δx and the boundary particle spacing Δp, namely $\beta = \Delta x / \Delta p$.

The parameter β ensures that if we change the spacing between the boundary particles, the force on the fluid is not changing.

If we assume that all the masses are equal ($m_k = m_a$), the above equation reduces to:

$$\boxed{\mathbf{f_{ak}} = \frac{K}{\beta} \frac{\mathbf{x_{ak}}}{|\mathbf{x_{ak}}|^2} W(|\mathbf{x_{ak}}|/h)} \tag{2.5}$$

We then proceed as in section 2.3.1 using equations (2.2) and (2.3).

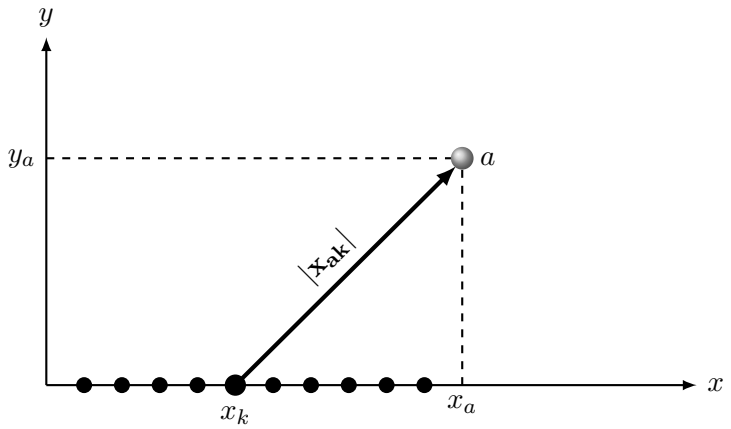


Figure 2.13: Radial boundary force approach

This approach removes the needing of computing the outward unit normals to the boundaries, which as we have seen was the origin of some ambiguities in the normal boundary force formulation. Moreover, it simplifies SPH algorithms and turns out to be superior to other formulations when dealing with complicated boundaries.

Chapter 3

Floating objects

The motion of a rigid body interacting with a fluid is determined by specifying the motion of the center of mass and the rotation about the center of mass (Monaghan, 2005; Gesteira et al., 2010). In figure 3.1, the vector **X** indicates the position of the center of mass, while **V** is the velocity of the center of mass.

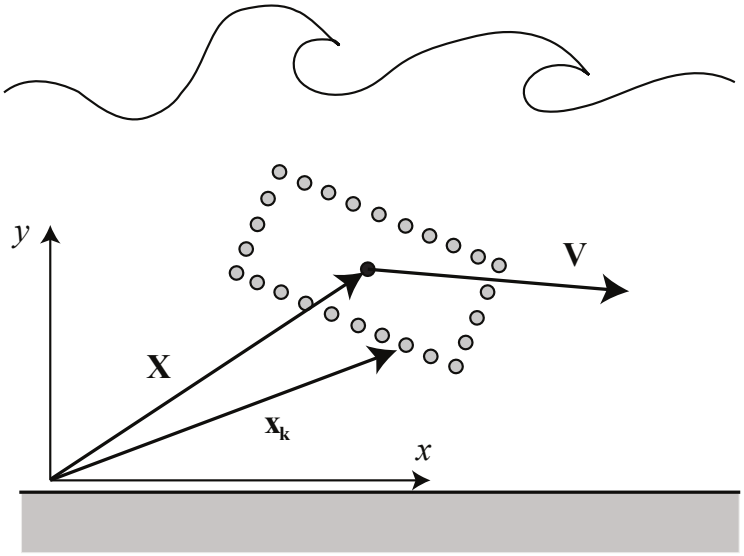


Figure 3.1: Floating object in a two-dimensional flow.

The rigid body is represented in the computation by a set of boundary particles that are equispaced around the boundary (Monaghan et al., 2003).

3.1 Equations of motion

The equations of motion for a floating object should describe the time evolution of the center of mass, the angular velocity and the position of the particles on the boundary of the rigid body.

Equations for the center of mass. The center of mass of the rigid body is evolved according to the following vector equation:

$$M \frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} = \sum_k m_k \, \mathbf{f_k} \tag{3.1}$$

where M represents the mass of the rigid body, the index k denotes the boundary particles on the surface of the rigid body, m_k is the mass of the boundary particles of the rigid

body and $\mathbf{f_k}$ are the forces acting on the boundary particles of the rigid body due to the surrounding fluid. For the formulation of $\mathbf{f_k}$ see section 2.3 and refer to figures 3.2 and 3.3, which sketch the concept of boundary forces calculation for a floating object by using the normal boundary force approach. Note that equation (3.1) takes care of the three translational degrees of freedom of the rigid object.

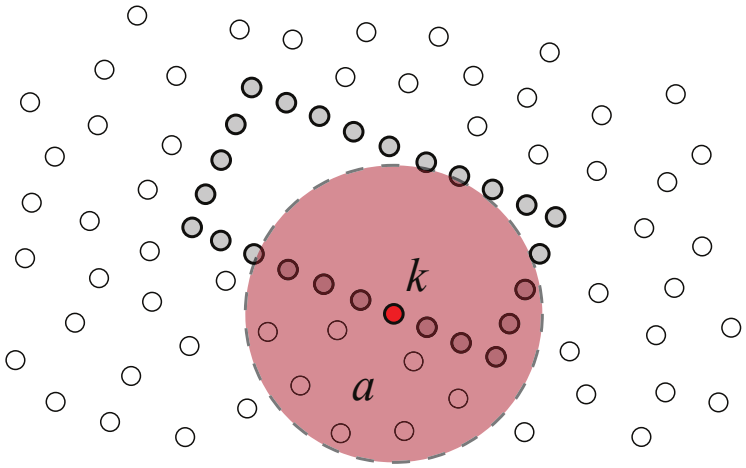


Figure 3.2: Calculation of the force on each boundary particle of the rigid body.

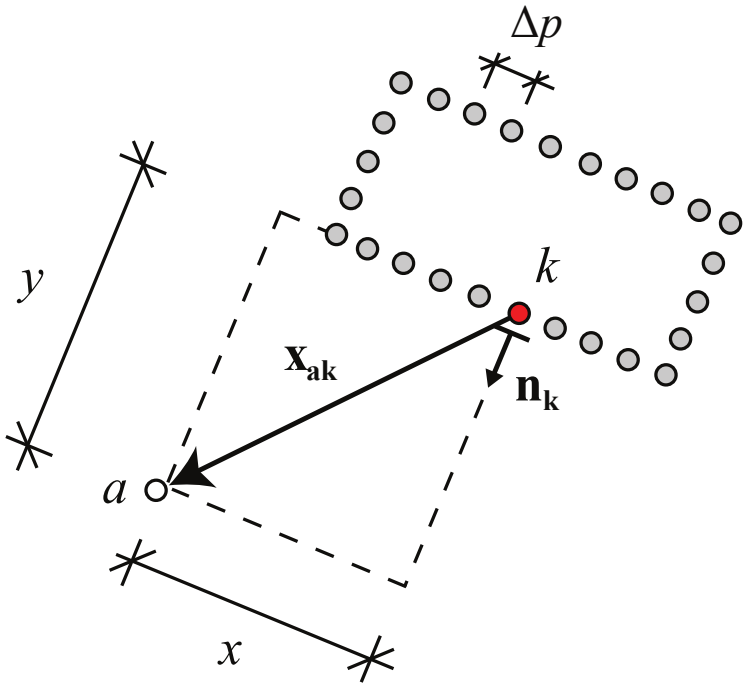


Figure 3.3: Calculation of the boundary force: force normal to the boundary.

Equations for the angular velocity. In the case of a two-dimensional motion, the equations for the evolution of the angular velocity $\mathbf{\Omega}$ are given by:

$$I \frac{\mathrm{d}\mathbf{\Omega}}{\mathrm{d}t} = \sum_k m_k(\mathbf{x_k} - \mathbf{X}) \times \mathbf{f_k} \tag{3.2}$$

where I is the moment of inertia and $\mathbf{x_k}$ denotes the position of the k-th boundary particle. The above equation takes care of the rotational degree of freedom of the rigid object. Note that $I\,\mathbf{\Omega}$ represents the *angular momentum* (also called moment of momentum).

Equations for the boundary particles. Finally, the boundary particles are moved according to:

$$\frac{\mathrm{d}\mathbf{x_k}}{\mathrm{d}t} = \mathbf{V} + \mathbf{\Omega} \times (\mathbf{x_k} - \mathbf{X}) \tag{3.3}$$

Equations (3.1), (3.2) and (3.3) are integrated in time to predict, respectively, the values of $\mathbf{V}$, $\mathbf{\Omega}$ and the position of the boundary particles for the beginning of the next time-step.

Now we should have a look at what happens to the SPH equations of motion when we introduced a floating object into the simulation. The SPH approximation of the equations of motion in presence of floating rigid objects becomes:

$$\frac{\mathrm{d}v_i^a}{\mathrm{d}t} = -\sum_{b=1}^{N} m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_i^a} + 2\sum_{b=1}^{N} m_b \left(\frac{\mu_a D_{ij}^{\prime a}}{\rho_a^2} + \frac{\mu_b D_{ij}^{\prime b}}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial x_j^a} + f_i^a$$

which is exactly the same equation that we get in section 2.3.1 when discussing *fixed* boundaries. So the boundary forces formulations that we discussed in the previous chapter are well suited for this kind of application.

Conclusions

We have seen that the boundary treatment is a critical aspect of the SPH method, and that it can be addressed by different approaches. In this project, we reviewed the most popular ones, trying to point out the advantages and disadvantages of each of them. For some of them we also showed concretely their key aspects by performing small implementations and carrying out simple simulations. It appeared that the choice of one method over another is often a trade-off between its pros and contras, and mostly a matter of practical experience and tuning of the parameters. Regardless of the precise form of boundary treatment that one might choose, we stressed that it can be easily included into the SPH approximation of equations of motion. We finally showed that this is also true for the interaction of the fluid with a floating rigid object.

Appendices

Appendix A

Smoothed particle hydrodynamics

The particle method is not only an approximation of the continuum fluid equations, but also gives the rigorous equations for a particle system which approximates the molecular system underlying the continuum equations.

— Von Neumann, 1944

In this appendix, a very short summary of the mathematics behind the SPH discretization technique is presented.

A.1 Integral representation of a function

Everything starts from the integral representation of a function, which can also be considered as a statement of one of the properties of the Dirac's delta function:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')\,\mathrm{d}\mathbf{x}'$$

If we replace Dirac's delta with a smoothing kernel function $W(\mathbf{x} - \mathbf{x}', h)$ we get:

$$f(\mathbf{x}) \doteq \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)\,\mathrm{d}\mathbf{x}'$$

This smoothing kernel function has the special property that it mimics Dirac's delta function when the smoothing length h approaches zero, namely:

$$\lim_{h\to 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}')$$

Once we have introduced the smoothing kernel $W(\mathbf{x} - \mathbf{x}', h)$, we can approximate the derivative of a function as:

$$< \boldsymbol{\nabla} \cdot f(\mathbf{x}) >= \int_{\Omega} \left[\boldsymbol{\nabla} \cdot f(\mathbf{x}')\right] W(\mathbf{x} - \mathbf{x}', h)\,\mathrm{d}\mathbf{x}'$$

where the symbols $< >$ are there just to remind us of the approximation that we are making.

A.2 SPH approximation of the value of a function

The SPH approximation of the value of a function at particle a is:

$$< f(\mathbf{x^a}) >= \sum_{b=1}^{N} \frac{m_b}{\rho_b} f(\mathbf{x^b})\, W_{ab}$$

The SPH approximation of the value of a function derivative at particle a is:

$$< \boldsymbol{\nabla} \cdot f(\mathbf{x^a}) >= \sum_{b=1}^{N} \frac{m_b}{\rho_b} f(\mathbf{x^b}) \cdot \boldsymbol{\nabla}_a W_{ab}$$

Figure A.1 depicts a two-dimensional domain in the x-y plane and shows a SPH kernel corresponding to a particle denoted as a. The SPH kernel has a compact support and a generic particle falling into the support domain of particle a is denoted as b. This notation is used throughout this report.

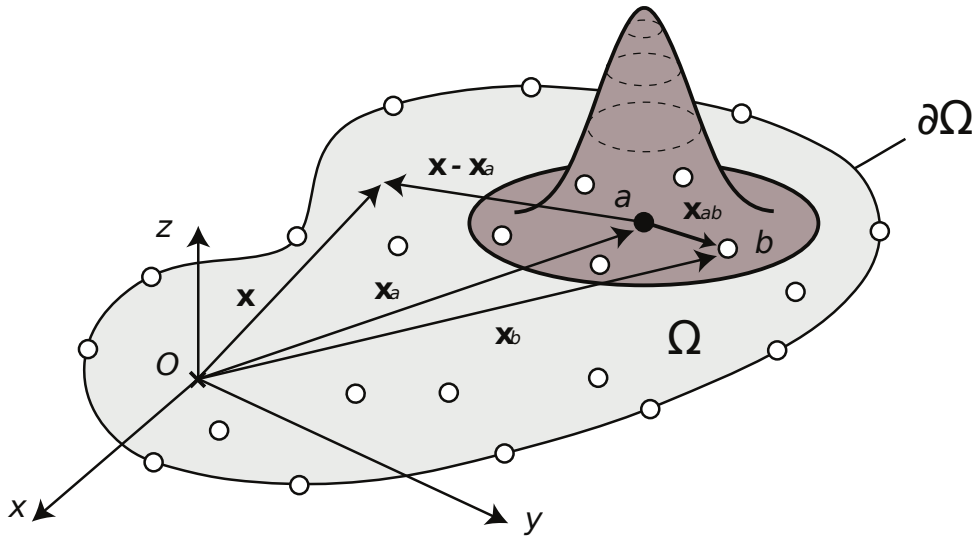


Figure A.1: Smoothed particle hydrodynamics.

Appendix B

Navier-Stokes equations

In this appendix, we will have a look at all the bricks that build up the Navier-Stokes equations, namely:

1. Continuity equation;
2. Constitutive equations for fluids;
3. Equations of motions;
4. Energy equation.

B.1 Continuity equation

The continuity equation is a statement of the *principle of conservation of mass*. It reads:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho \boldsymbol{\nabla} \cdot \mathbf{v} \tag{B.1}$$

Since $\boldsymbol{\nabla} \cdot \mathbf{v}$ represents the time rate of change of the volume per unit volume (or *volumetric strain*):

$$\boldsymbol{\nabla} \cdot \mathbf{v} = \frac{1}{\delta V} \frac{\mathrm{D}(\delta V)}{\mathrm{D}t}$$

the continuity equation (B.1) is telling us that the volume can change only if ρ can vary in time, i.e. only if the fluid is compressible.

If the fluid is incompressible, ρ is constant for all times, so that the continuity equation (B.1) reduces to:

$$\boldsymbol{\nabla} \cdot \mathbf{v} = 0 \tag{B.2}$$

This version of the continuity equation for incompressible fluids takes the name of *incompressibility condition*.

B.2 Constitutive equations for fluids

In order to describe the kinematics of a fluid flow, we define the *velocity gradient* as (Malvern, 1977; Romano et al., 2006):

$$\boldsymbol{\nabla}\mathbf{v} = \mathbf{D} + \mathbf{W}$$

where $\mathbf{D}$ is the symmetric part of $\boldsymbol{\nabla}\mathbf{v}$ and it is known as the *rate of deformation* or *stretching tensor* (Euler, 1770):

$$\mathbf{D} = \mathrm{sym}(\boldsymbol{\nabla}\mathbf{v}) = \frac{\boldsymbol{\nabla}\mathbf{v} + \mathbf{v}\boldsymbol{\nabla}}{2}$$

Let's remark that the first invariant of $\mathbf{D}$ is equal to its trace, which in turns gives the velocity divergence:

$$I_D = \mathrm{tr}(\mathbf{D}) = \boldsymbol{\nabla} \cdot \mathbf{v}$$

Moreover, let's remember that the *deviator* of $\mathbf{D}$ is defined as:

$$\mathbf{D}' = \mathbf{D} - \frac{1}{3}(\mathrm{tr}\mathbf{D})\mathbf{I} \tag{B.3}$$

The deviator of the rate of deformation tensor will come up often in the later discussion.

The *spin* or *vorticity* tensor $\mathbf{W}$ is defined as the skew-symmetric part of the velocity gradient:

$$\mathbf{W} = \mathrm{skew}(\boldsymbol{\nabla}\mathbf{v}) = \frac{\boldsymbol{\nabla}\mathbf{v} - \mathbf{v}\boldsymbol{\nabla}}{2}$$

In the following, we will review three different formulations of the constitutive equations for fluids.

B.2.1 Navier-Poisson law of compressible viscous fluids

The Navier-Poisson law of compressible viscous fluids (also known as Navier-Stokes behaviour) relates the stress tensor to the rate of deformation tensor according to the following equation (Malvern, 1977; Romano et al., 2006):

$$\mathbf{T} = -p(\rho)\mathbf{I} + \lambda(\rho)(\mathrm{tr}\mathbf{D})\mathbf{I} + 2\mu(\rho)\mathbf{D} \tag{B.4}$$

The first term in this relationship denotes the static component of the pressure; the second term indicates the dynamic component of the pressure, since it depends on $\mathrm{tr}(\mathbf{D})$ which we have seen to be equal to the velocity divergence $\boldsymbol{\nabla} \cdot \mathbf{v}$; the third term represents the shear stress. Briefly:

$$\mathbf{T} = \underbrace{-p(\rho)\mathbf{I}}_{\substack{\text{static}\\ \text{component}\\ \text{of the pressure}}} + \underbrace{\lambda(\rho)(\mathrm{tr}\mathbf{D})\mathbf{I}}_{\substack{\text{dynamic}\\ \text{component}\\ \text{of the pressure}}} + \underbrace{2\mu(\rho)\mathbf{D}}_{\substack{\text{shear}\\ \text{stresses}}}$$

The constants λ and μ are two independent parameters characterizing the elastic behaviour of a given body; they were introduced by Gabriel Lamé and thus are often referred to as *Lamé constants*. They have to be determined for every material by means of experiments.

B.2.2 Compressible viscous fluids with no bulk viscosity

The *bulk modulus* (also known as *modulus of compression* or *bulk viscosity*) is defined as (Malvern, 1977):

$$k = \lambda + \frac{2}{3}\mu$$

The condition of zero bulk viscosity ($k = 0$, also known as *Stokes condition*) implies:

$$\lambda = -\frac{2}{3}\mu \tag{B.5}$$

Under this condition, equation (B.4) becomes:

$$\begin{aligned}\mathbf{T} &= -p\mathbf{I} - \frac{2}{3}\mu(\mathrm{tr}\mathbf{D})\mathbf{I} + 2\mu\mathbf{D} \\ &= -p\mathbf{I} + 2\mu\left[\mathbf{D} - \frac{1}{3}(\mathrm{tr}\mathbf{D})\mathbf{I}\right]\end{aligned}$$

Finally, remembering equation (B.3), we get:

$$\mathbf{T} = \underbrace{-p\mathbf{I}}_{\substack{\text{isotropic}\\ \text{part of } \mathbf{T}}} + \underbrace{2\mu\mathbf{D}'}_{\substack{\text{deviatoric}\\ \text{part of } \mathbf{T}}} \tag{B.6}$$

This equation governs the behaviour of compressible viscous fluids with no bulk viscosity. The first term in the equation, namely $-p\mathbf{I}$, is the isotropic part of the stress tensor. The second term is the deviatoric part of the stress tensor, $2\mu\mathbf{D}'$, which is called *shear stress tensor* and is denoted as $\mathbf{T}'$. In fact, $\mathbf{T}'$ is clearly related to shear since $\mathbf{D}'$ contains the angular deformations.

B.2.3 Incompressible viscous fluids

To obtain the constitutive equation for incompressible viscous fluid it is sufficient to introduce in equation (B.6) the *incompressibility condition* $\boldsymbol{\nabla} \cdot \mathbf{v} = 0$ (equation (B.2)). Note that, under such condition, $\mathbf{D}' \equiv \mathbf{D}$ and therefore equation (B.6) becomes

$$\mathbf{T} = \underbrace{-p\mathbf{I}}_{\substack{\text{isotropic}\\ \text{part of } \mathbf{T}}} + \underbrace{2\mu\mathbf{D}}_{\substack{\text{deviatoric}\\ \text{part of } \mathbf{T}}} \tag{B.7}$$

B.3 Equations of motion

The equations of motion for a continuous medium are derived from the *momentum principle* or *Newton's second law* and they have the general form (Romano et al., 2006):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \boldsymbol{\nabla} \cdot \mathbf{T} + \rho\mathbf{b} \tag{B.8}$$

These equations are also known as *Cauchy's equations of motion* (Romano et al., 2006). Of course, the specific form of the equation of motion (B.8) depends on the constitutive relationship we choose for the stress tensor $\mathbf{T}$, as we will see in the following subsections.

B.3.1 Compressible fluids with bulk viscosity

If we specialize equation (B.8) by using the constitutive equation (B.4), we get (Romano et al., 2006):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla}(\lambda \boldsymbol{\nabla} \cdot \mathbf{v}) + \boldsymbol{\nabla} \cdot (2\mu\mathbf{D}) + \rho\mathbf{b}$$

Recalling that $2\mathbf{D} = \boldsymbol{\nabla}\mathbf{v} + \mathbf{v}\boldsymbol{\nabla}$, the previous equation can also be rewritten as:

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla}(\lambda \boldsymbol{\nabla} \cdot \mathbf{v}) + \boldsymbol{\nabla} \cdot (\mu\boldsymbol{\nabla}\mathbf{v} + \mu\mathbf{v}\boldsymbol{\nabla}) + \rho\mathbf{b}$$

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla}(\lambda \boldsymbol{\nabla} \cdot \mathbf{v}) + \mu\nabla^2\mathbf{v} + (\mu\boldsymbol{\nabla} \cdot \mathbf{v})\boldsymbol{\nabla} + \rho\mathbf{b}$$

Finally we get (Malvern, 1977):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + (\lambda + \mu)\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{v}) + \mu\nabla^2\mathbf{v} + \rho\mathbf{b} \tag{B.9}$$

B.3.2 Compressible fluids with no bulk viscosity

To obtain the equations of motion for compressible fluids with no bulk viscosity it is sufficient to introduce the Stokes condition (B.5) into equation (B.9) and get (Malvern, 1977):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \frac{1}{3}\mu \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{v}) + \mu \nabla^2 \mathbf{v} + \rho \mathbf{b} \tag{B.10}$$

The above equation can be obtained also from equation (B.8) by plugging into it the constitutive equation for a compressible fluid with no bulk viscosity (equation (B.6)).

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \boldsymbol{\nabla} \cdot (-p\mathbf{I} + 2\mu \mathbf{D}') + \rho \mathbf{b}$$

This also provides an alternative expression of the equations of motion, namely:

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \mathbf{T}' + \rho \mathbf{b} \tag{B.11}$$

where $\mathbf{T}' = 2\mu \mathbf{D}'$ is the shear stress tensor.

B.3.3 Incompressible flows

It is sufficient to plug the incompressibility condition (B.2) into equation (B.10) and get (Romano et al., 2006):

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{b} \tag{B.12}$$

B.4 Incompressible vs compressible flow analysis

Let us consider a three-dimensional incompressible flow at constant temperature. In such a case, we have to find four unknowns in order to completely describe the fluid flow: the pressure p and the three components of velocity $\mathbf{v}$. **In an incompressible flow analysis we need four equations**: the incompressibility condition $\boldsymbol{\nabla} \cdot \mathbf{v} = 0$ and the three scalar equations of motion from the vector equation (B.12). These equations are coupled since the velocity $\mathbf{v}$ appears in both of them (Çengel and Cimbala, 2010). The pressure and the velocity are also coupled since they both appear in equation (B.12). When the velocity is known, if we integrate the pressure gradient appearing in equation (B.12), we can compute the pressure field up to an arbitrary constant. To determine this arbitrary constant we need to measure the pressure somewhere in the field, i.e. we need a boundary condition on the pressure. Most of the codes in computational fluid dynamics do not calculate pressure by direct integration of the Navier-Stokes equations, but the use a pressure correction algorithm based on a form of *Poisson's equation* for Δp (Çengel and Cimbala, 2010):

$$\nabla^2(\Delta p) = RHS(n) \tag{B.13}$$

Note that since pressure appears only as a gradient in the incompressible Navier-Stokes equations, the absolute magnitude of pressure is not important, only pressure differences matter.

This is not true for compressible flow, where p is the thermodynamic pressure rather than the mechanical one. To find p in compressible flows, we need to introduce an *equation of state*, which relates the pressure, temperature and density of the fluid (see section B.6). Since the equation of state introduces **temperature** as an additional unknown, we also need an additional equation, which expresses the *conservation of energy principle* (see section B.5). **Therefore, in a compressible flow analysis, we have to deal with**

six equations in six unknowns. For simple compressible systems, the total energy consists of internal, kinetic and potential energy, expressed on a unit-mass basis:

$$e = u + k_e + p_e = u + \frac{1}{2}V^2 + gz$$

The internal energy is the sum of all microscopic forms of energy (energy related to the molecular structure of a system and the degree of molecular activity, also referred to as thermal energy). The kinetic energy is the energy that a system possesses as a result of its motion. The potential energy is the energy that a system possesses as a result of its elevation in a gravitational field.

B.5 Energy equation

The energy equation is the mathematical formulation of the *first law of thermodynamics*, which expresses the *conservation of energy*:

> *The time rate of change of the energy inside an infinitesimal fluid cell equals the summation of the heat flux into that fluid cell, and the time rate of work done by the body and surface forces acting on that fluid cell.*

The general form of the energy equation, neglecting heat flux term and the body force, is (Romano et al., 2006):

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = \mathbf{T} : \boldsymbol{\nabla}\mathbf{v} \tag{B.14}$$

Here, e is a specific energy per unit mass [J/kg] and : denotes the double-dot product (or scalar product) between two tensors. In indicial notation, equation (B.14) can be written as:

$$\rho\, e_{,t} = T_{ij} v_{i,j}$$

with $i, j = 1, 2, 3$.

If we consider a compressible viscous fluid with no bulk viscosity, we can use the constitutive law (B.6) to obtain:

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = (-p\mathbf{I} + 2\mu\mathbf{D}') : \boldsymbol{\nabla}\mathbf{v}$$

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = -p(\boldsymbol{\nabla}\cdot\mathbf{v}) + 2\mu(\mathbf{D}' : \boldsymbol{\nabla}\mathbf{v}) \tag{B.15}$$

This equation states that the time rate of change of the specific internal energy is equal to the work done by the isotropic pressure multiplying the volumetric strain plus the energy dissipation due to the viscous shear forces:

$$\rho \underbrace{\frac{\mathrm{d}e}{\mathrm{d}t}}_{\substack{\text{time rate of change of the specific internal energy}}} = \underbrace{-p(\boldsymbol{\nabla}\cdot\mathbf{v})}_{\substack{\text{work done by the isotropic pressure multiplying the volumetric strain}}} + \underbrace{2\mu(\mathbf{D}' : \boldsymbol{\nabla}\mathbf{v})}_{\substack{\text{energy dissipation due to the viscous shear forces}}}$$

B.6 Equation of state

An equation of state is any equation that relates pressure, temperature and density of a substance (Çengel and Cimbala, 2010).

B.6.1 Equation of state for gas

The most famous equation of state is the ideal gas law:

$$p = \rho R T$$

where p is the absolute pressure, ρ is the density, R is the gas constant and T is the absolute temperature. In the SPH calculations, we will use the so-called *gamma law*:

$$p = (\gamma - 1)\rho e$$

with $\gamma = 1.4$.

B.6.2 Equation of state for artificial water

The water in SPH is treated as weakly compressible, thus we will refer to this kind of fluid as *artificial water*. This allows the use of an equation of state to determine fluid pressure, which is much faster than solving Poisson's equation (B.13) that we discussed in section B.4 (Gesteira et al., 2010). Usually, **the compressibility is adjusted so that the time-step of the simulation is reasonable**. To approximate the real fluid as an artificial compressible fluid, a smaller value than the actual sound speed should be used so that the time-step is increased to an acceptable value; but it should also be large enough so that the behaviour of the artificial compressible fluid is sufficiently close to the reality (Liu and Liu, 2003).

The equation of state most frequently used when atmospheric pressure is negligible is (Monaghan, 2005):

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

where ρ_0 is the reference density, $\gamma \cong 7$ and B is a parameter chosen so that the speed of sound c_s is large enough to limit the relative density variation $|\delta\rho|/\rho$ (i.e., to endow the fluid with a slight compressibility in order to permit the use of an equation of state).

The sound speed c_s is determined according to the *Newton-Laplace formula* for the speed of pressure waves:

$$c_s = \sqrt{\frac{K}{\rho_0}} \tag{B.16}$$

where K is the *bulk modulus*, which measures the substance's resistance to uniform compression. For a gas, $K = \gamma p$, so that equation (B.16) becomes:

$$c_s = \sqrt{\frac{\gamma p}{\rho_0}} \tag{B.17}$$

Usually the sound speed in water at 25°C is about 1497 m/s for freshwater and 1560 m/s for seawater. If we choose $p = B$ and work out B from equation (B.17), we get:

$$B = \rho_0 \frac{c_s^2}{\gamma} \tag{B.18}$$

The relative density variation is related to the *Mach number*:

$$\frac{|\delta\rho|}{\rho} \sim M^2$$

where $M = v/c_s$, with v being the maximum fluid speed. Therefore we can ensure a slight density variation, say $|\delta\rho|/\rho \sim 0.01$, if $M < 0.1$, namely if $c_s \sim 10v$. This means that the

sound speed should be about ten times faster than the maximum fluid velocity in order to keep density variations within less than 1% (Gesteira et al., 2010).

In fact, if we substitute $c_s = 10v$ into equation (B.18), we get:

$$B = 100\rho_0 \frac{v^2}{\gamma}$$

and the relative density variation is ~ 0.01.

As can be seen, the above formulas require an estimation of the maximum fluid speed. For instance, when a dam of height H collapses, an adequate value for the maximum water velocity is $v^2 = 2gH$ (Monaghan, 1994; Monaghan et al., 2003) or $v^2 = gH$ (Monaghan and Kos, 1999).

Bibliography

Gesteira, M., Rogers, B., Dalrymple, R., Crespo, A., and Narayanaswamy, M. (2010). *User Guide for the SPHysics code.*

Liu, G. and Liu, M. (2003). *Smoothed Particle Hydrodynamics: A Meshfree Particle Method.* World Scientific, Singapore.

Malvern, L. E. (1977). *Introduction to the Mechanics of a Continuous Medium.* Prentice Hall.

Monaghan, J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2):399 – 406.

Monaghan, J. J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759.

Monaghan, J. J. and Kajtar, J. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180:1811–1820.

Monaghan, J. J. and Kos, A. (1999). Solitary waves on a cretan beach. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 125(3):145–154.

Monaghan, J. J., Kos, A., and Issa, N. (2003). Fluid motion generated by impact. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 129:250–259.

Randles, P. and Libersky, L. (1996). Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):375–408.

Romano, A., Lancellotta, R., and Marasco, A. (2006). *Continuum Mechanics using Mathematica - Fundamentals, Applications, and Scientific Computing.* Birkhäuser.

Çengel, Y. A. and Cimbala, J. M. (2010). *Fluid Mechanics: Fundamentals and Applications - second edition in SI units.* McGraw-Hill.