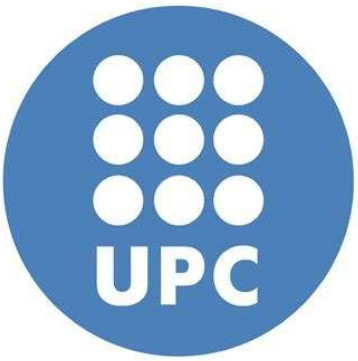
Double Master's Degree Program



Università degli Studi di Padova

Dipartimento di Ingegneria Civile, Edile e Ambientale

Universitat Politècnica de Catalunya

Departament d'Enginyeria Civil i Ambiental

Smoothed Particle Hydrodynamics (SPH): towards implicit time integration

---

Candidate:
Pilotto Marco

Student number:
2090112

Supervisors:
Prof. Javier Bonet
Prof. Riccardo Rossi
Prof. Antonia Larese

Academic year 2024/2025

A chi crede in me

Abstract

This thesis investigates the smoothed particle hydrodynamics (SPH) method, which is a numerical mesh-less method particularly suited for the simulation of free surface flows. The formulation is based on a total Lagrangian description of a system of first-order conservation laws, expressed in terms of the linear momentum and the Jacobian of the deformation. In this framework, the evaluation of spatial integrals is performed with respect to the initial undeformed configuration, resulting in an efficient approach that completely circumvents the need for continuous particle neighbor searching. To ensure stability and consistency at the SPH discretisation level, a characteristic-based Riemann solver is employed, designed to preserve both the accuracy and the conservation properties of the overall algorithm.

This work presents a novel implicit formulation of the Smoothed Particle Hydrodynamics (SPH) method. The chosen time integration scheme is the second-order Crank–Nicolson method, while the nonlinearities arising from the implicit formulation are handled through a Newton–Raphson iterative procedure. Unlike traditional explicit schemes, the proposed method removes the restrictions on the time step size imposed by the Courant–Friedrichs-Lewy (CFL) condition, enabling simulations over longer time scales and expanding the range of applications. The new formulation is implemented and validated through a set of benchmark tests.

Contents

Chapter 1

Introduction

Smoothed particle hydrodynamics (SPH) is a mesh-less method originally developed by Monaghan for astrophysical applications [23]. This method is commonly adopted in engineering and fluid dynamics to model free surface flows [30] due to its ability to handle complex surfaces and large deformations.
The most attractive feature of SPH is its mesh-free nature, which allows for the discretization of a continuum into a set of particles without the necessity of a background computational grid, allowing to manage highly deformable materials. The interaction between particles is based entirely on their reciprocal distance. In general, this method results reasonable accurate and competitive in terms of computational cost, with respect to other methods such as finite volume (FVM).
In this work, as in [22], a Lagrangian formulation is adopted, where the particle variables are updated based on the initial configuration of the continuum. This approach allows neighbor searches, essential for weighted averaging, to be performed only once at the beginning of the simulation, reducing computational costs. However, this simplification limits the applicability of the method in scenarios involving significant topological changes in the free surface, such as impact simulations, where the reference to the initial configuration becomes meaningless due to changes in particle neighborhoods.
The evolution of the continuum is governed by two fundamental conservation equations: the conservation of volume and the conservation of linear momentum. These equations, written in terms of the determinant of the deformation gradient J and the first Piola-Kirchhoff stress tensor, are essential for accurately modeling compressible and incompressible flows.
This thesis considers a constitutive material law that accounts only for volumetric deformation, and the shear component is neglected. This constitutive model is appropriate for simulating fluids, which typically exhibit limited shear resistance.
As noted in [19], the meshless nature of SPH introduces inherent instabilities that requires careful numerical treatment to ensure stability and accuracy. To this end, a Riemann-based upwinding stabilization is introduced. This approach effectively handles shocks and suppresses spurious modes. Moreover, it enables localized and appropriately scaled stabilization, improving both robustness and accuracy.

Objective and motivations

In the literature, only explicit time integration schemes are typically used in SPH simulations. Although they are easy to implement, these schemes need to satisfy the Courant-Friedrichs-Lewy (CFL) condition, which imposes severe restrictions on the time step, significantly limiting computational efficiency. To overcome this limitation, an implicit time

integration scheme is proposed, allowing for larger time steps while maintaining accuracy. The proposed scheme is based on Crank-Nicolson combined with Newton-Raphson iterations to correctly handle the material and geometric nonlinearities. This allows the time step to be independent of the material properties and the number of particles used in the simulation, enabling longer simulation times and expanding the range of application to new scenarios.

Thesis Structure

This thesis is structured as follows. Chapter 1 introduces the SPH method, detailing its formulation, interpolation functions, kernel approximation techniques, and kernel correction methods. Chapter 2 describes the physical problem under investigation, presenting the governing equations, the material constitutive law and their SPH discretization. Chapter 3 focuses on time integration strategies, presenting the novel implicit alternative. In Chapter 4, a simple one-dimensional tension test is conducted on an elastic material, with results from both explicit and implicit schemes compared in terms of accuracy and performance. Finally, Chapter 5 extends the analysis to multidimensional problems, detailing the derivation of the residuals required for Newton-Raphson implementation and presenting a set of benchmark cases to compare the accuracy and efficiency of the proposed implicit method with the original explicit formulation.

Chapter 2

Introduction to Smoothed Particle Hydrodynamics (SPH)

This chapter aims to provide an introduction to the SPH method and show how a continuous field, such as pressure or velocity, can be approximated correctly through a discrete set of particles that carry the field properties [23, 24].

2.1 Discrete approximation of a continuous field

To begin, recall the integral representation of a scalar function via the Dirac delta

$$f(x) = \int_V f(x')\ \delta(x - x')dx' \tag{2.1}$$

where $f(x)$ is a scalar field, $\delta(x-x')$ is the Dirac delta function and x' is a dummy variable ranging over the volume V. In practice, equation (2.1) states that the value of a function at one point can be obtained as the integral of the function itself over its domain, weighted by the Dirac delta distribution. In the SPH formulation a generalization of the δ-function is introduced. The so-called smoothing kernel $W(x)$ with a finite support controlled by a smoothing length h. This kernel satisfies the consistency condition

$$lim_{h\to 0} W(x - x', h) = \delta(x - x') \tag{2.2}$$

The smoothing kernel is subject to normalization, i.e. the integral of $W(x - x', h)$ over V is unitary. The kernel function can substitute the Dirac δ-function and, as demonstrated in Remark 2, equation (2.1) can be equivalently expressed as

$$f(x) = \int_V \frac{f(x')}{\rho(x')}\ W(x - x', h)\ \rho(x')dx' + \mathcal{O}(h^2) \tag{2.3}$$

Finally, by discretizing (2.3) into a series of particles with mass $m_b = \rho(x_b)dx_b$, the identity (2.1) becomes

$$f(x) \approx \sum_{b\in\Lambda_a} \frac{m_b}{\rho_b} f(x_b)\ W(x - x_b, h) \tag{2.4}$$

The equation (2.4) represent the discrete approximation of function f at any point x of the domain V. Note that the values of $f(x)$ are interpolated at the position of the particles, which allows to express the values of a continuous function $f(x_a)$ as a weighted sum over the neighboring particles. Indeed, each particle interacts with all the other particles that

are within the support of the kernel function, this distance h is the so-called smoothing length.

$$b \in \Lambda_a \iff |x_a - x_b| \leq h \tag{2.5}$$

The interaction of each particle and its neighbors is weighted by the kernel function $W(x_a - x_b, h)$ itself, where x_a is the position of particle a and x_b the position of its neighbor b.
Figure 2.1 shows the dependency of the kernel function on inter-particle distance, highlighting its role in weighting the influence of each neighbor on the estimation of the field variable in the particle a. Recall that the modulation of each contribution depends both on distance and mass m_b.

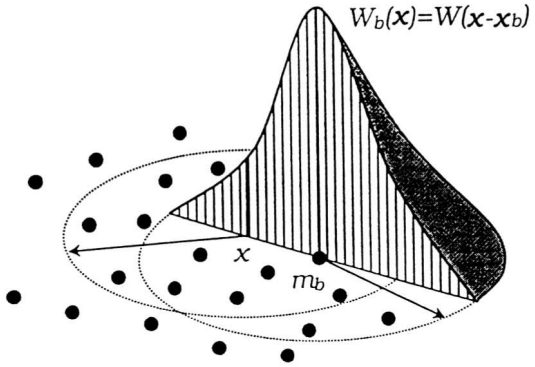


Figure 2.1: Particle interpolation and kernel function [3]

In continuum mechanics, it is often not sufficient to approximate only the value of a function itself. Indeed, functions derivatives play an important role in describing phenomena such as deformation and stress. In particular, the spatial derivative of a scalar field can be derived using equation (2.4)

$$\frac{d}{dx} f(x) \approx \sum_{b \in \Lambda_a} \frac{m_b}{\rho_b} f(x_b) \; \frac{d}{dx} W(x - x_b, h) \tag{2.6}$$

Note that the only part that depends on x is the smoothing kernel W, consequently, to compute the gradient of a continuous field approximated with SPH, the derivative of the field $f(x)$ is not needed. This is very useful because kernels are often simpler and their derivatives can be easily precomputed.

Remark 1. Similar identities may be given for a vector field $\boldsymbol{f}(\boldsymbol{x})$. Indeed,

$$\boldsymbol{f}(\boldsymbol{x}_a) \approx \sum_{b \in \Lambda_a} \frac{m_b}{\rho_b} \boldsymbol{f}(\boldsymbol{x}_b) \; W(\boldsymbol{x} - \boldsymbol{x}_b, h) \tag{2.7}$$

When computing the divergence of $\boldsymbol{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, only the smoothing kernel depends on $\boldsymbol{x}$. Thus, the discretized form is

$$\nabla \cdot \boldsymbol{f}(\boldsymbol{x}_a) \approx \sum_{b \in \Lambda_a} \frac{m_b}{\rho_b} \boldsymbol{f}(\boldsymbol{x}_b) \; \cdot \nabla W(\boldsymbol{x} - \boldsymbol{x}_b, h) \tag{2.8}$$

While taking the gradient of $\boldsymbol{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, the following expression is obtained

$$\nabla \boldsymbol{f}(\boldsymbol{x}_a) \approx \sum_{b \in \Lambda_a} \frac{m_b}{\rho_b} \boldsymbol{f}(\boldsymbol{x}_b) \; \otimes \nabla W(\boldsymbol{x} - \boldsymbol{x}_b, h) \tag{2.9}$$

In SPH, in equations (2.7) and (2.8) is often added the term $-\boldsymbol{f}(\boldsymbol{x}_a)$, this term assure to correctly evaluate the gradient, or divergence, of a constant vector field. In this work an approximation for the Laplacian of a function $\Delta\boldsymbol{f}(\boldsymbol{x})$ is not required, but in principle it can be obtained by applying the Laplacian operator directly to the smoothing kernel $W(\boldsymbol{x})$.

Remark 2. The Taylor expansion on $f(x')$ is performed and substituted into (2.1) to obtain

$$f(x) = \int_V \frac{f(x')}{\rho(x')}\, W(x-x',h)\, \rho(x')dx' + \int_V \frac{f'(x')}{\rho(x')}\, (x-x')\, W(x-x',h)\, \rho(x')dx' + \mathcal{O}(h^2)$$

now note that the second integral cancels out for symmetric kernels.

2.2 Smoothing kernels

In the previous section, the discretisation of a continuous field was introduced through the use of specific functions, known as smoothing kernels $W(\boldsymbol{x})$. These functions are selected carefully, as they must satisfy certain properties. In particular, the kernel function must have compact support, meaning that only a limited set of neighboring particles contributes to the approximation of the field at particle a.
Additionally, equation (2.2) must be satisfied (i.e. the kernel tends to the Dirac δ-function as $h \to 0$). The integral of the kernel over its support must be equal to one, ensuring correct normalization. The discrete version of the latter condition reads as

$$\sum_{b\in\Lambda_a} V_b W_b(\boldsymbol{x}_a) = 1 \qquad \text{with} \quad W_b(\boldsymbol{x}_a) = W(\boldsymbol{x}_a - \boldsymbol{x}_b) \tag{2.10}$$

where V_b is the volume associated to particle b which act as a weight.
Recall equations (2.8) or (2.9), the computation of the gradient or divergence of a vector field $\boldsymbol{f}(\boldsymbol{x})$ requires the computation of the derivatives of $W(\boldsymbol{x})$. Consequently, a suitable smoothing kernel must have a continuous and well-defined first derivative. Moreover, since the derivative of a continuous field is the null vector, the following condition is introduced

$$\sum_{b\in\Lambda_a} V_b \nabla W_b(\boldsymbol{x}_a) = 0 \tag{2.11}$$

Finally, the kernel functions should be spherically symmetric, and thus depends only on h and $\boldsymbol{x} - \boldsymbol{x}_b$.
The latter property would naturally suggest to use a Gaussian-like smoothing kernel, such as

$$W(\boldsymbol{r},h) = \frac{1}{h^3\pi^{\frac{3}{2}}} \exp^{-\left(\frac{r}{h}\right)^2} \tag{2.12}$$

where $\boldsymbol{r} = \boldsymbol{x}_a - \boldsymbol{x}_b$. However, this function has the drawback to have as support the whole domain ($W > 0 \;\forall\; a \in \Omega$), this means that all the particles contribute as neighbors. For a large number of particles the computational cost of using a Gaussian kernel is prohibitive considering also that the long-range interactions or contributions are negligible.
Thus, it makes sense to look for smoothing kernels $W(\boldsymbol{x})$ with compact support to speed up the computation.
In the literature, various types of kernels are presented and utilized to solve SPH problems, for example, the quadratic kernel [16] or the linear kernel [4].

This work uses the Cubic Spline kernel described in [1] that satisfies all the requirements. The Cubic Spline kernel is defined as

$$W(q) = \alpha_d \begin{cases} \frac{2}{3} - q^2 + \frac{1}{2}q^3 & 0 \leq q < 1 \\ \frac{1}{6}(2-q)^3 & 1 \leq q < 2 \\ 0 & q \geq 2 \end{cases} \tag{2.13}$$

where $q = \frac{\|\boldsymbol{r}\|}{h}$ and α_d are the kernel normalization factors for the respective problem dimensions ($\alpha_1 = \frac{1}{h}$, $\alpha_2 = \frac{15}{7\pi h^2}$ and $\alpha_3 = \frac{3}{2\pi h^3}$).
The gradient with respect to $\boldsymbol{r}$ of the kernel function in equation (2.12) is computed, using the chain rule, as

$$\nabla W(q) = \frac{\partial W(q)}{\partial \boldsymbol{r}} = \frac{\partial W(q)}{\partial q}\frac{\partial q}{\partial \boldsymbol{r}} = \frac{\partial W(q)}{\partial q}\frac{\boldsymbol{r}}{h \parallel \boldsymbol{r} \parallel} \tag{2.14}$$

The 1-dimensional Cubic Spline kernel is plotted together with its derivative in figure 2.2. The symmetry of the kernel and the continuity and regularity of its derivative are shown.

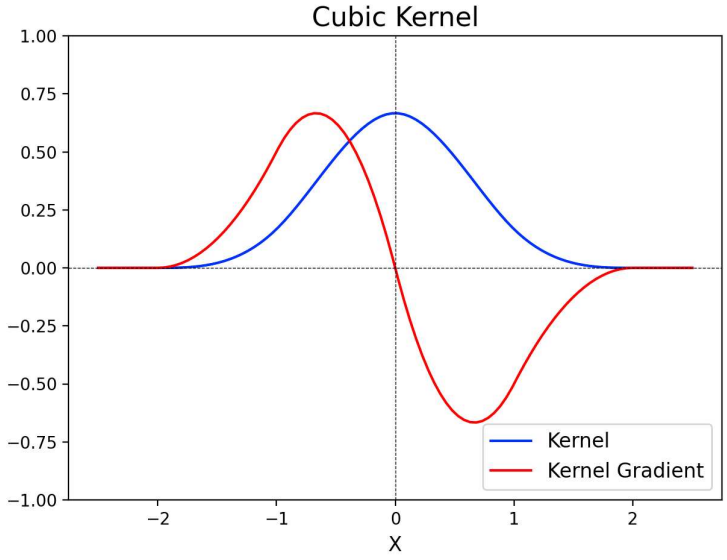


Figure 2.2: Cubic Spline Kernel and Cubic Spline kernel derivative in 1D. In this case the smoothing length is $h = 1$, so that the kernel's support is $[-2h, 2h]$.

2.3 Preservation of linear and angular momentum

It is known that a motion of a continuum must preserve linear and angular momentum in the absence of external forces; this is represented by a set of two equations, namely one for mass conservation and another one for the conservation of linear momentum. The system of equations can be derived by a variational formulation that consists of the minimization of a potential function. In particle discretization, factors such as non-uniform particle distributions or the truncation of the support domain near boundaries can lead to a lack of momentum conservation. This section examines the conditions under which a system of differential equations, discretized using Smoothed Particle Hydrodynamics (SPH), satisfies these conservation laws, following the methodology presented in [3]. The conservation of mass is written in terms of continuity equation, introducing a density field such that

$$\dot{\rho}_a = -\rho_a \, \nabla \cdot \boldsymbol{v}_a \tag{2.15}$$

then, the SPH approximation of the velocity field reads as

$$\nabla \cdot \boldsymbol{v}_a = \sum_{b \in \Lambda_a} V_b \ \boldsymbol{v}_b \ \cdot \nabla W_b(\boldsymbol{x}_a) \tag{2.16}$$

Moreover, the motion of a set of particles obeys the conservation of linear momentum, which, in absence of dissipative effects, can be written as

$$m_a \dot{\boldsymbol{v}}_a = \boldsymbol{F}_a - \boldsymbol{T}_a \tag{2.17}$$

where $\boldsymbol{F}_a$ and $\boldsymbol{T}_a$ represent the external and internal forces acting on particle a, respectively. The internal forces are computed as the sum of the interaction forces between one particle and its set of neighbors

$$\boldsymbol{T}_a = \sum_{b \in \Lambda_a} \boldsymbol{T}_{ab} \tag{2.18}$$

where $\boldsymbol{T}_{ab}$ is expressed as

$$\boldsymbol{T}_{ab} = V_b \ p_b \mathbb{I} \cdot \nabla W_a(\boldsymbol{X}_b) \tag{2.19}$$

where p represents the pressure field. This expression for the internal forces considers, for simplicity, an isotropic material[1] and it is introduced just with the purpose of carrying out the following proofs. A more detailed definition of the framework adopted in this work for internal forces will be discussed in the next chapter.
The volume ratio between the initial and current state is denoted by $J = V/V_0 = \rho_0/\rho$. Thus, it is possible to define an internal energy function per unit of undeformed volume U depending on J, such that

$$p = \frac{dU}{dJ} \tag{2.20}$$

Therefore, when a continuum is represented by a set of particles, the total potential energy functional can be represented as

$$\Pi(\boldsymbol{x}) = \sum_{a}^{N} V_a U(J) \tag{2.21}$$

where the potential energy function $U(J)$ depends on J the volume variation. For example $U(J) = \frac{1}{2}\kappa(J-1)^2$. At this point, it is useful, to recall that the variation, or directional derivative, of a general functional Ψ [5] is defined as

$$D\Psi(\delta \boldsymbol{v}) = \left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0} \Psi(\boldsymbol{x} + \varepsilon \delta \boldsymbol{v}) \tag{2.22}$$

The variation of equation (2.21), in absence of external forces, leads to the following expression, where the directional derivative is expressed using internal forces

$$D\Pi[\delta \boldsymbol{v}] = \sum_{a}^{N} \boldsymbol{T_a} \cdot \delta \boldsymbol{v}; \qquad \frac{\partial \Pi}{\partial \boldsymbol{x}_a} = \boldsymbol{T}_a \tag{2.23}$$

Remark 3. This remark aims to demonstrate the validity of equation (2.23), which is equivalent to showing that the SPH formulation of the problem can be derived from a variational principle. Recalling equation (2.20) and using chain rule, the directional derivative of the potential energy is computed as

$$D\Pi[\delta \boldsymbol{v}] = \sum_{a=1}^{N} V_a DU(J)[\delta \boldsymbol{v}] = \sum_{a=1}^{N} V_a p_a \left(-\frac{\rho_a^0}{\rho_a^2} \right) D\rho_a[\delta \boldsymbol{v}] = -\sum_{a=1}^{N} V_a \left(\frac{p_a \rho_a^0}{\rho_a^2} \right) D\rho_a[\delta \boldsymbol{v}] \tag{2.24}$$

[1] Then, the Chauchy stess tensor can be expressed in term of pressure as $\sigma = p\mathbb{I}$.

Since the continuity equation (2.15) is used to update the density, the directional derivative of the density field reads as

$$D\rho_a[\delta \boldsymbol{v}] = -\rho_a \ \nabla \cdot \delta \boldsymbol{v} = -\rho_a \sum_{b \in \Lambda_a} V_b \ \delta \boldsymbol{v}_b \cdot \nabla W_b(\boldsymbol{x}_a) \tag{2.25}$$

where the velocity $\boldsymbol{v}$ is replaced by the virtual velocity $\delta \boldsymbol{v}$. Substituting this equation in (2.24) and recalling that by construction $\nabla W_b(\boldsymbol{x}_a) = -\nabla W_a(\boldsymbol{x}_b)$, the expression (2.24), once the indices are inverted, becomes

$$D\Pi[\delta \boldsymbol{v}] = \sum_{a,b}^{N} V_a V_b \ p_a \mathbb{I} \cdot \nabla W_b \left(\boldsymbol{x}_a\right) \cdot (-\delta \boldsymbol{v}_b) = \sum_{a}^{N} V_a \left[\sum_{b \in \Lambda_a}^{N} V_b \ p_b \mathbb{I} \cdot \nabla W_a \left(\boldsymbol{x}_b\right) \right] \cdot \delta \boldsymbol{v}_a \tag{2.26}$$

by simply comparing this last expression with equation (2.19), the proof is completed.

Preservation of linear momentum
The linear momentum of the system is computed as

$$\boldsymbol{p} = \sum_{a=1}^{N} m_a \boldsymbol{v}_a \tag{2.27}$$

then the linear momentum is conserved if its time derivative satisfies the following expression

$$\dot{\boldsymbol{p}} = \sum_{a=1}^{N} m_a \dot{\boldsymbol{v}}_a = -\sum_{a=1}^{N} \boldsymbol{T}_a = 0 \tag{2.28}$$

the equivalence derive from equation (2.17). Considering (2.17) and recalling that $\nabla W_a(\boldsymbol{X}_b) = -\nabla W_b(\boldsymbol{X}_a)$, it is clear that $\boldsymbol{T}_{ab} = -\boldsymbol{T}_{ba}$ and consequently the sum of all internal forces will vanish.
Since the set of internal forces $\boldsymbol{T}_a$ is derived from a potential function, it is possible to prove the conservation of linear momentum by showing the invariance of the potential function with respect to rigid body translation. Consider the variation of the potential energy Π due to an arbitrary uniform velocity field $\boldsymbol{v}_0$, knowing that the potential energy is invariant under rigid body translation the following expression must be satisfied

$$D\Pi(\delta \boldsymbol{v}_0) = \sum_{a=1}^{N} \boldsymbol{T}_a \cdot \boldsymbol{v}_0 = 0 \tag{2.29}$$

since $\boldsymbol{v}_0$ is arbitrary, then the only possibility is that the set of internal forces, represented by $\boldsymbol{T}_a$, sums to zero. Thus, if the internal forces are derived from a potential function that is invariant with respect to rigid body translations, the sum of internal forces will necessarily be zero, ensuring linear momentum conservation.

Preservation of angular momentum
The angular momentum of the system is computed as

$$\boldsymbol{L} = \sum_{a=1}^{N} x_a \times m_a \boldsymbol{v}_a \tag{2.30}$$

then the angular momentum is conserved if its time derivative satisfies the following expression

$$\dot{\boldsymbol{L}} = \sum_{a=1}^{N} x_a \times m_a \dot{\boldsymbol{v}}_a = -\sum_{a=1}^{N} x_a \times \boldsymbol{T}_a = 0 \tag{2.31}$$

The total moment of interacting forces between two particles with respect to the origin is computed as

$$\boldsymbol{x}_a \times \boldsymbol{T}_{ab} + \boldsymbol{x}_b \times \boldsymbol{T}_{ba} = -(\boldsymbol{x}_b - \boldsymbol{x}_a) \times \boldsymbol{T}_{ab} \tag{2.32}$$

where the two terms have been contracted in one using the property $\boldsymbol{T}_{ab} = -\boldsymbol{T}_{ba}$. Expression (2.26) vanishes if and only if $\boldsymbol{T}_{ab}$ is parallel to the vector $(\boldsymbol{x}_b - \boldsymbol{x}_a)$. Equation (2.14) tells that the kernel gradient is proportional (parallel) to the vectorial distance between the two particles, this means that (2.31) will vanish if and only if the stresses are isotropic. Generally, materials do not present an isotropic behavior, this means that a more general derivation is needed to show conservation of angular momentum.
To demonstrate the conservation of angular momentum, consider the invariance of the potential energy $\Pi(\boldsymbol{x})$ with respect to a rigid body rotation about the origin. The velocity variation due to an infinitesimal rotation is given by

$$\delta\boldsymbol{v}_a = \delta\boldsymbol{w} \times \boldsymbol{x}_a \tag{2.33}$$

where $\delta\boldsymbol{w}$ is the angular velocity vector. The invariance of $\Pi(\boldsymbol{x})$ under this transformation implies

$$0 = D\Pi[\delta\boldsymbol{w} \times \boldsymbol{x}_a] = \sum_{a=1}^{N} \boldsymbol{T}_a \cdot (\delta\boldsymbol{w} \times \boldsymbol{x}_a) = \delta\boldsymbol{w} \cdot \sum_{a=1}^{N} \boldsymbol{x}_a \times \boldsymbol{T}_a \tag{2.34}$$

by arbitrariness of $\delta\boldsymbol{w}$ the total internal moment vanishes[2], ensuring the conservation of angular momentum.
To complete the proof, it remains to verify that the SPH approximation correctly preserves a rigid body rotation. Consider a rigid body motion characterized by the angular velocity vector $\boldsymbol{w} = [w_x,\ w_y,\ w_z]^T$. The velocity vector at an arbitrary point $\boldsymbol{x}$ is computed as

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{w} \times \boldsymbol{x} \tag{2.35}$$

the gradient of this velocity field is found by taking the derivative with respect to $\boldsymbol{x}$ of equation (2.35)

$$\nabla\boldsymbol{v} = \boldsymbol{W}; \qquad \boldsymbol{W} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{2.36}$$

Note that the divergence (or trace) of the deformation tensor $\boldsymbol{\epsilon} = (\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)/2$ is equal to the null vector given the skew symmetric nature of $\boldsymbol{W}$. Then, in absence of deformation, the total potential energy of the system would remain constant proving its invariance with respect to rigid body motions. This is valid in absence of numerical approximations, but in SPH framework the velocity gradient is evaluated as

$$\nabla\boldsymbol{v} = \sum_{b\in\Lambda_a} V_b(\boldsymbol{W}\boldsymbol{x}_b - \boldsymbol{W}\boldsymbol{x}_a) \otimes \nabla W_b(\boldsymbol{x_a}) = \boldsymbol{W}\left(\sum_{b\in\Lambda_a} V_b(\boldsymbol{x}_b - \boldsymbol{x}_a) \otimes \nabla W_b(\boldsymbol{x_a})\right) \tag{2.37}$$

$\nabla\boldsymbol{v} \neq \boldsymbol{W}$ implies that the SPH approximation does not perfectly represent a rigid body rotation, and consequently, angular momentum is not conserved.
The correct skew tensor $\boldsymbol{W}$ is obtained if and only if the following condition is satisfied

$$\sum_{b\in\Lambda_a} V_b(\boldsymbol{x}_b - \boldsymbol{x}_a) \otimes \nabla W_b(\boldsymbol{x_a}) = \mathbb{I} \qquad \text{for } a = 1, \ldots, N \tag{2.38}$$

Condition (2.33) is not satisfied in standard SPH algorithms and therefore angular momentum is not preserved. To overcome this problem, an appropriate correction needs to be applied on the kernel gradient.

[2]The vector identity $a \cdot (b \times c) = b \cdot (c \times a)$ has been used

2.4 Corrected kernel and corrected gradient

To fulfill condition (2.38) to ensure the conservation of angular momentum, different types of correction can be adopted [3]. A mixed kernel and gradient kernel correction is presented in this section and adopted in this work.
Recalling (2.4), an arbitrary function $\boldsymbol{f}(\boldsymbol{x})$ can be approximated in the SPH context using a corrected kernel

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{b=1}^{N} V_b \boldsymbol{f}_b \tilde{W}_b(\boldsymbol{x}) \quad \text{with} \quad \tilde{W}_b(\boldsymbol{x}) = \frac{W_b(\boldsymbol{x})}{\sum_{b=1}^{N} V_b W_b(\boldsymbol{x})} \tag{2.39}$$

where the corrected kernel $\tilde{W}(\boldsymbol{x})$ is obtained by normalizing the kernel itself assuring the satisfaction of equation (2.10) by construction. The gradient of $\boldsymbol{f}(\boldsymbol{x})$ is evaluated as

$$\nabla \boldsymbol{f}(\boldsymbol{x}) = \sum_{b=1}^{N} V_b \boldsymbol{f}_b \otimes \tilde{\nabla} \tilde{W}_b(\boldsymbol{x}_a) \tag{2.40}$$

where the "corrected gradient of a corrected kernel" is computed as:

$$\tilde{\nabla} \tilde{W}_b(\boldsymbol{x}_a) = \boldsymbol{L}_a \nabla \tilde{W}_b(\boldsymbol{x}_a) \tag{2.41}$$

The so called gradient of a corrected kernel in equation (2.41) is obtained by taking the derivative with respect to $\boldsymbol{x}$ of (2.39b)

$$\nabla \tilde{W}_b(\boldsymbol{x}) = \frac{\nabla W_b(\boldsymbol{x}) \sum_{b=1}^{N} V_b W_b(\boldsymbol{x}) - W_b(\boldsymbol{x}) \sum_{b=1}^{N} V_b \nabla W_b(\boldsymbol{x})}{\left(\sum_{b=1}^{N} V_b W_b(\boldsymbol{x}) \right)^2} \tag{2.42}$$

While the correction matrix $\boldsymbol{L}_a$ ensure that equation (2.38) is satisfied by construction.

$$\boldsymbol{L}_a = \left(\sum_{b=1}^{N} V_b \nabla \tilde{W}_b(\boldsymbol{x}_a) \otimes (\boldsymbol{x}_b - \boldsymbol{x}_a) \right)^{-1} \tag{2.43}$$

To conclude, after selecting an appropriate kernel function, it is necessary to apply these corrections. From now on, all the kernels and kernel gradients defined will consider these corrections.

Chapter 3

Total Lagrangian SPH formulation

In this chapter, a system of two Lagrangian conservation laws, respectively for linear momentum and volume, is presented. These two equations describe the deformation of the continuum in time. Then, the system is spatially discretized in space within the corrected SPH framework. Finally, an adapted Riemann-based stabilization technique is described to mitigate the numerical instabilities.

3.1 Total Lagrangian first order conservation laws

3.1.1 Conservation of volume

Consider the three-dimensional deformation of an elastic body moving from its initial configuration, occupying a volume Ω_0 with boundary $\partial\Omega_0$ and outward unit normal $\boldsymbol{N}$, to a current configuration at time t occupying a volume Ω, with boundary $\partial\Omega$ and outward unit normal $\boldsymbol{n}$.

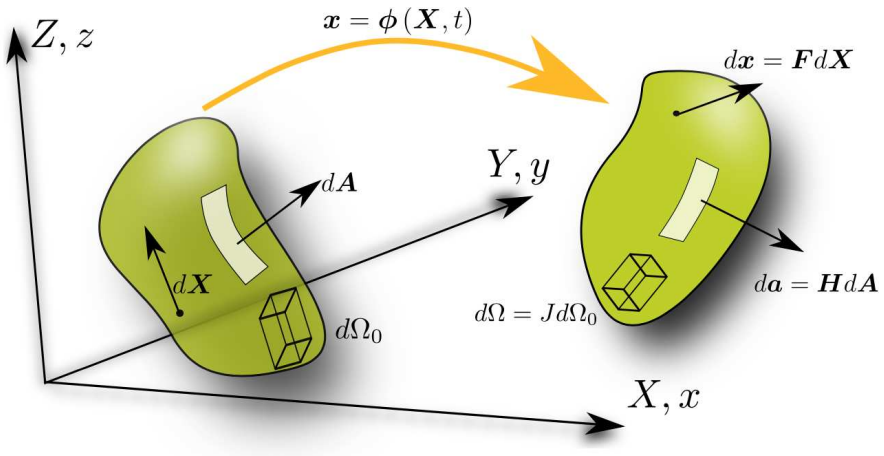


Figure 3.1: Motion of a deformable continuum domain [22]

The deformation is defined by a map $\boldsymbol{x} = \phi(\boldsymbol{X}, t)$ which satisfies the volume and linear momentum conservation laws. The notation for the deformation gradient $\boldsymbol{F}$ (that maps $d\boldsymbol{x} = \boldsymbol{F}d\boldsymbol{X}$) and its determinant J (that maps $d\Omega = Jd\Omega_0$) are

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \mathbf{X}} = \nabla_0 \boldsymbol{x} \qquad J = \det(\boldsymbol{F}) \tag{3.1}$$

in which $\boldsymbol{x}$ represents the current position of a particle initially at position $\mathbf{X}$, while ∇_0 denotes the gradient with respect to the initial configuration. To map the material

differential area vectors $d\boldsymbol{A}$ from the initial to the current configuration $d\boldsymbol{a}$, the co-factor of deformation $\boldsymbol{H}$ is introduced

$$\boldsymbol{H} = J\boldsymbol{F}^{-T} \tag{3.2}$$

The relation between the co-factor and the deformation gradient is given by the Nanson formula [9]. Using the co-factor of deformation is possible to express the Jacobian of the deformation via the following integral conservation law

$$\frac{d}{dt}\int_{\Omega_0} J d\Omega_0 = \int_{\partial\Omega_0} (\boldsymbol{H}^T\boldsymbol{v})\cdot\boldsymbol{N} dA \tag{3.3}$$

where $\boldsymbol{v}$ is the velocity field. Once the divergence theorem is applied on (3.3), the equivalent local conservation law can be written as

$$\frac{\partial J}{\partial t} - \mathrm{DIV}(\boldsymbol{H}^T\boldsymbol{v}) = 0 \qquad c[\![J]\!] = -[\![\boldsymbol{v}]\!]\cdot(\boldsymbol{H}^{Ave}\boldsymbol{N}) \tag{3.4}$$

together with the associated jump condition across a discontinuity surface [18] with normal $\boldsymbol{N}$ and propagating with velocity c.
In this equation, DIV represents the material divergence operator, $\boldsymbol{H}^{Ave} = \frac{1}{2}(\boldsymbol{H}^+ + \boldsymbol{H}^-)$ is an average of the co-factor of deformation between the left and right side of the discontinuity surface and $[\![\cdot]\!] = [\cdot]^+ - [\cdot]^-$ represents the jump operator.

Remark 4. An equivalent non-conservative version [1] of (3.4) is obtained by introducing the Piola identity (DIV $\boldsymbol{H} = 0$) to give

$$\frac{\partial J}{\partial t} - \boldsymbol{H} : \nabla_0\boldsymbol{v} = 0 \tag{3.5}$$

3.1.2 Conservation of linear momentum

The conservation of linear momentum per unit of undeformed volume is expressed, for any arbitrary Lagrangian volume Ω_0, as

$$\frac{d}{dt}\int_{\Omega_0} \rho_0\ \boldsymbol{v}\ d\Omega_0 = \int_{\Omega_0} \boldsymbol{b}\ d\Omega_0 + \int_{\partial\Omega_0} \boldsymbol{t}\ dA \tag{3.6}$$

where $\boldsymbol{b}$ is a body force per unit of volume and $\boldsymbol{t} = \boldsymbol{P}\boldsymbol{N}$ is the traction vector associated with outward normal surface vector and with the first Piola-Kirchhoff stress tensor. Thus, the local conservation law and corresponding jump condition becomes

$$\rho_0\frac{\partial \boldsymbol{v}}{\partial t} - \mathrm{DIV}\boldsymbol{P} = \boldsymbol{b} \qquad c\rho_0[\![\boldsymbol{v}]\!] = -[\![\boldsymbol{P}]\!]\boldsymbol{N} \tag{3.7}$$

3.1.3 Combined equations

Combining equations (3.7) and (3.5), the system of total Lagrangian conservation laws can be summarized [18, 19, 20] as

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}_I}}{\partial \boldsymbol{X_I}} = \boldsymbol{\mathcal{S}} \quad \forall I = 1,2,3 \tag{3.8}$$

[1] The right hand side is not expressed as divergence of a flux.

where $\boldsymbol{\mathcal{U}}$ represents the set of conservation variables, $\boldsymbol{\mathcal{S}}$ the source term and $\boldsymbol{\mathcal{F}_I}$ the flux vector in the material direction I.

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \rho_0 \boldsymbol{v} \\ J \end{bmatrix} \quad \boldsymbol{\mathcal{F}_I} = - \begin{bmatrix} \boldsymbol{P E_I} \\ \boldsymbol{H} : (\boldsymbol{v} \otimes \boldsymbol{E_I}) \end{bmatrix} \quad \boldsymbol{\mathcal{S}} = \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} \tag{3.9}$$

with $\boldsymbol{E_I}$ is the I-th unit vector of the Cartesian basis[2].
The first Piola-Kirchhoff stress tensor allows to relate the stress in the current (deformed) configuration to forces per unit area defined in the reference configuration [5]. In general, it is defined as

$$\boldsymbol{P} = J \boldsymbol{\sigma} \boldsymbol{F}^{-T} \tag{3.10}$$

The system of equations (3.8) needs the introduction of a constitutive law relating the first Piola stress tensor to J, acting as a closure. To do so, the following elastic strain energy per unit of undeformed volume is introduced

$$\psi(J) = \frac{1}{2} \kappa (J-1)^2 \tag{3.11}$$

where κ is the bulk modulus. In this work, only an elastic volumetric contribution will be considered. Then, equation (3.10) reduces to

$$\boldsymbol{P} = -p\boldsymbol{H}; \qquad p = \frac{d\psi}{dJ} = \kappa(J-1) \tag{3.12}$$

where a negative value for pressure indicates compression. To simulate a more generic situation, an additional inelastic effect $\boldsymbol{P}^v$ should also be taken into account.

$$\boldsymbol{P} = -p\boldsymbol{H} + \boldsymbol{P}^v \tag{3.13}$$

these effects in general derive from viscous effects and more complex constitutive laws needs to be considered. Physically, this implies that no shear/deviatoric stresses, viscosity or heat conduction are considered. In this work, these effects are neglected, so that $\boldsymbol{P}^v = 0$.
Finally, the appropriate initial conditions must be specified as well as the boundary conditions to fully define the initial boundary value problem.

3.2 Weak formulation

In general, a standard weak variational statement [19] for the system in equation (3.8) is obtained by multiplying the local form of the conservation laws by an appropriate work conjugate virtual field $\delta\boldsymbol{\mathcal{V}} := \{\delta\boldsymbol{v}, \delta\Sigma_J\}$, and integrating over the domain Ω_0, to give

$$\int_{\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} d\Omega_0 = \int_{\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \boldsymbol{\mathcal{S}} d\Omega_0 \; - \int_{\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \frac{\partial \boldsymbol{\mathcal{F}_I}}{\partial \boldsymbol{X_I}} d\Omega_0 \tag{3.14}$$

the symbol $\bullet$ denote an inner dual product of work conjugate pairs. The application of the Gauss-Green divergence theorem on equation (3.14) leads to

$$\int_{\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} d\Omega_0 = \int_{\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \boldsymbol{\mathcal{S}} d\Omega_0 \; + \int_{\Omega_0} \boldsymbol{\mathcal{F}_I} \bullet \frac{\partial \delta \boldsymbol{V}}{\partial X_I} d\Omega_0 \; - \int_{\partial\Omega_0} \delta\boldsymbol{\mathcal{V}} \bullet \boldsymbol{\mathcal{F}_N} dA \tag{3.15}$$

[2] $\boldsymbol{E_1} = [1,0,0]$, $\boldsymbol{E_2} = [0,1,0]$ and $\boldsymbol{E_3} = [0,0,1]$

where $\boldsymbol{\mathcal{F}_N} = \boldsymbol{\mathcal{F}_I} N_I$ are the normal fluxes with N_I as the material outward normal in the I-th direction. The expression (3.15) can be written in extended form as

$$\begin{aligned} \int_{\Omega_0} \delta\boldsymbol{v} \cdot \rho \frac{\partial \boldsymbol{v}}{\partial t} d\Omega_0 &= \underbrace{-\int_{\Omega_0} \boldsymbol{P} : \nabla_0 \delta\boldsymbol{v} \; d\Omega_0}_{\delta\mathcal{W}_{int}} + \underbrace{\int_{\Omega_0} \delta\boldsymbol{v} \cdot \boldsymbol{b} \; d\Omega_0 + \int_{\Omega_0} \delta\boldsymbol{v} \cdot \boldsymbol{t} \; dA}_{\delta\mathcal{W}_{ext}} \\ \int_{\Omega_0} \delta\Sigma_J \frac{\partial J}{\partial t} d\Omega_0 &= \int_{\Omega_0} \delta\Sigma_J \; (\boldsymbol{H} : \nabla_0 \boldsymbol{v}) \; d\Omega_0 \end{aligned} \tag{3.16}$$

By introducing the Hamiltonian $\mathcal{H} = \mathcal{H}(\boldsymbol{X}, t)$ it is possible to give a physical meaning to the conjugate virtual field $\delta\boldsymbol{\mathcal{V}}$. Indeed, the Hamiltonian is defined as

$$\mathcal{H}(\boldsymbol{X}, t) = \hat{\mathcal{H}}(\boldsymbol{\mathcal{U}}) = \frac{1}{2}\boldsymbol{v} \cdot \boldsymbol{v} + \psi(J) \tag{3.17}$$

that is the summation of the kinetic energy and the Helmholtz free energy [3] per unit of undeformed volume. The work conjugate are obtained by differentiating the above expression

$$\boldsymbol{\mathcal{V}} = \frac{\partial \hat{\mathcal{H}}}{\partial \boldsymbol{\mathcal{U}}} = \begin{bmatrix} \frac{\partial \hat{\mathcal{H}}}{\partial \boldsymbol{v}} \\ \frac{\partial \hat{\mathcal{H}}}{\partial J} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} \tag{3.18}$$

Hence, $\delta\boldsymbol{v}$ and $\delta\sum_J$ represent virtual velocity and virtual pressure fields respectively. To show that the system of equation proposed is coherent and physically admissible, the Hamiltonian is considered in terms of the second law of thermodynamics. In particular, the time derivative of the Hamiltonian can be written as

$$\begin{aligned} \frac{d}{dt}\int_{\Omega_0} \mathcal{H} d\Omega_0 &= \int_{\Omega_0} \boldsymbol{\mathcal{V}} \; \bullet \; \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} d\Omega_0 \\ &= \int_{\Omega_0} \left(\boldsymbol{v} \cdot \rho_0 \frac{\partial \boldsymbol{v}}{\partial t} - p \frac{\partial J}{\partial t} \right) \; d\Omega_0 \\ &= \int_{\Omega_0} \left(\boldsymbol{v} \cdot \rho_0 \frac{\partial \boldsymbol{v}}{\partial t} - p\boldsymbol{H} : \nabla_0 \boldsymbol{v} \right) \; d\Omega_0 \\ &= \int_{\Omega_0} \left(\boldsymbol{v} \cdot \rho_0 \frac{\partial \boldsymbol{v}}{\partial t} - (\boldsymbol{P} - \boldsymbol{P}^v) : \nabla_0 \boldsymbol{v} \right) \; d\Omega_0 \end{aligned} \tag{3.19}$$

where equations (3.5) and (3.13) have been substituted in the second and third line respectively. At this point also the linear momentum equation (3.7) is substituted to give

$$\frac{d}{dt}\int_{\Omega_0} \mathcal{H} d\Omega_0 = \int_{\Omega_0} (\boldsymbol{v} \cdot \boldsymbol{b}_0 + \boldsymbol{v} \cdot \mathrm{DIV}\boldsymbol{P} + \boldsymbol{P} : \nabla_0 \boldsymbol{v}) \; d\Omega_0 \; - \int_{\Omega_0} \boldsymbol{P}^v : \nabla_0 \boldsymbol{v} \; d\Omega_0 \tag{3.20}$$

consider the identity $\boldsymbol{v} \cdot \mathrm{DIV}\boldsymbol{P} + \boldsymbol{P} : \nabla_0 \boldsymbol{v} = \mathrm{DIV}(\boldsymbol{P}^T \boldsymbol{v})$, then the above equation becomes

$$\frac{d}{dt}\int_{\Omega_0} \mathcal{H} d\Omega_0 = \int_{\Omega_0} \left(\boldsymbol{v} \cdot \boldsymbol{b}_0 + \mathrm{DIV}(\boldsymbol{P}^T \boldsymbol{v}) \right) \; d\Omega_0 \; - \int_{\Omega_0} \boldsymbol{P}^v : \nabla_0 \boldsymbol{v} \; d\Omega_0 \tag{3.21}$$

By applying the divergence theorem to the DIV term in (3.21), it yields

$$\frac{d}{dt}\int_{\Omega_0} \mathcal{H} d\Omega_0 - \underbrace{\int_{\Omega_0} \boldsymbol{v} \cdot \boldsymbol{b}_0 \; d\Omega_0 \; - \int_{\partial\Omega_0} \boldsymbol{v} \cdot \boldsymbol{t}_B \; dA}_{\dot{\Pi}_{ext}} = -\int_{\Omega_0} \boldsymbol{P}^v : \nabla_0 \boldsymbol{v} \; d\Omega_0 \tag{3.22}$$

[3] Notice that in an isothermal process the Helmholtz free energy is equivalent to the internal energy.

where $\dot{\Pi}_{ext}$ denotes the power introduced by external forces. In equation (3.22), the right-hand side represents the possible physical dissipation given by the viscous term and is always non positive (i.e. ≤ 0). Consequently, the equation is expressed as the following inequality

$$\frac{d}{dt}\int_{\Omega_0} \mathcal{H} d\Omega_0 - \dot{\Pi}_{ext} \leq 0 \tag{3.23}$$

This inequality tells that, in absence of external forces, the energy of the system is either conserved or decreased due to dissipation introduced by inelastic effects. Satisfaction of this inequality, as established through the Coleman-Noll procedure [13], is a necessary condition to guarantee the stability of the system.

3.3 Spatial discretization

In Smoothed Particles Hydrodynamics (SPH) the problem variables $\boldsymbol{\mathcal{U}}$ and the conjugate pairs $\delta\boldsymbol{\mathcal{V}}$ are interpolated via SPH kernel functions $W(\boldsymbol{X})$. In particular, the variables are interpolated at each particle as a weighted sum of the values from neighboring particles located within the compact support of the kernel function (Figure 3.2).

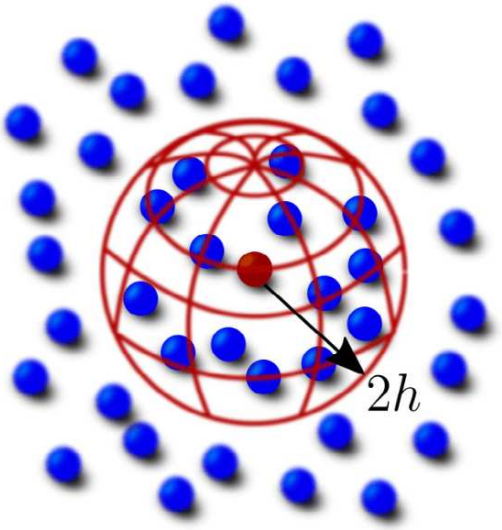


Figure 3.2: Particle approximation [22]

Finally, for a given particle initially at position $\boldsymbol{X}_a$,[4] both $\boldsymbol{\mathcal{U}}$ and $\delta\boldsymbol{\mathcal{V}}$ can be approximated, at any time, as

$$\boldsymbol{\mathcal{U}}_a(t) \approx \sum_{b\in\Lambda_a} V_b\, \boldsymbol{\mathcal{U}}_b(t)\, W_b(\boldsymbol{X}_a) \qquad \delta\boldsymbol{\mathcal{V}}_a \approx \sum_{b\in\Lambda_a} V_b\, \delta\boldsymbol{\mathcal{V}}_b\, W_b(\boldsymbol{X}_a) \tag{3.24}$$

in this equation, Λ_a represents the set of neighboring particles that lies inside the kernel support while V_b represents the volume associated with each particle b.
In this work, in order to ensure correct interpolation of both constant and linear functions, the correction of both gradient $\tilde{\nabla}_0(\bullet)$ and kernel $\tilde{W}(\boldsymbol{X})$ is applied as shown in section 2.4. The resulting approximation for an arbitrary function $\boldsymbol{f}(\boldsymbol{X})$ is

$$\boldsymbol{f}(\boldsymbol{X}_a) = \sum_{b\in\Lambda_a} V_b \boldsymbol{f}_b \tilde{W}_b(\boldsymbol{X}_a) \tag{3.25}$$

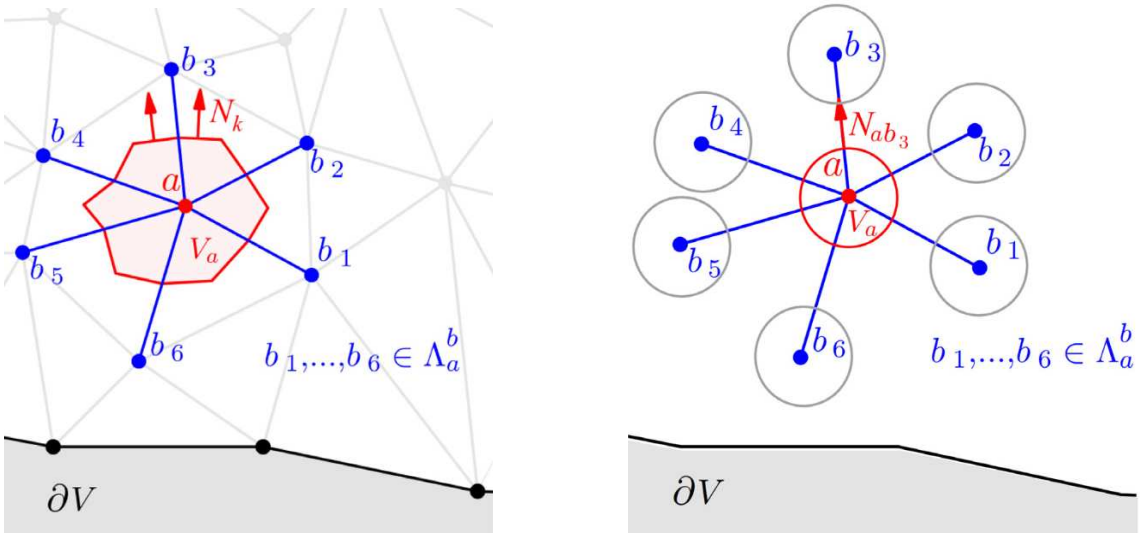
[4] Capital letter $\boldsymbol{X}$ refers to the initial configuration while $\boldsymbol{x}$ to the current one

while its gradient becomes

$$\nabla_0 \boldsymbol{f}(\boldsymbol{X}_a) = \sum_{b \in \Lambda_a} V_b \boldsymbol{f}_b \otimes \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a) \tag{3.26}$$

Remark 5. In this remark, an important relation connecting the Finite Volume Method (FVM) and the Smoothed Particle Hydrodynamics (SPH) is presented. This relation will be crucial in reformulating the Riemann-based FVM upwinding stabilization [21] within the SPH context.
It is known that another equivalent representation for the kernel gradient can be obtained by artificially adding the redundant term $\pm \boldsymbol{f}_a$ in (3.26)

$$\nabla_0 \boldsymbol{f}(\boldsymbol{X}_a) = \sum_{b \in \Lambda_a} V_b \, 2 \left[\frac{1}{2} (\boldsymbol{f}_b \pm \boldsymbol{f}_a) \right] \otimes \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a) \tag{3.27}$$



(a) (b)

Figure 3.3: Control volume for (a) vertex centred FVM and (b) SPH mesh free approach in 2D. The red area is the control volume associated to node a while the blue lines represents the edges connecting node or particle a to his neighbors. The vertex centered FVM requires the outward unit normal $\boldsymbol{N}_{ab}$ to be defined using the material outward normal $\boldsymbol{N}_k$ of every facet k belonging to edge ab. However, for the SPH mesh free approach, the normal $\boldsymbol{N}_{ab}$ is defined by a direction vector between particles a and b.[19]

FVM requires the definition of a dual mesh which is constructed by connecting edge midpoints, element centroids, and face centroids ensuring that each dual mesh element contains only a single node (see figure 3.3)[2, 11]. The discrete gradient at node a can be evaluated over its associated control volume by applying a discrete form of the Green–Gauss divergence theorem

$$\nabla_0 \boldsymbol{f}(\boldsymbol{X}_a) \approx \frac{1}{V_a} \int_{\partial V_a} \boldsymbol{f} \otimes \boldsymbol{N} dA \tag{3.28}$$

where $\boldsymbol{N}$ is the outward normal and V_a the control volume. The above integral (3.28) can be discretized by means of the FVM adopting a central difference scheme for the vector function $\boldsymbol{f}$ as

$$\nabla_0 \boldsymbol{f}(\boldsymbol{X}_a) \approx \frac{1}{V_a} \sum_{b \in \Lambda_a} \left[\frac{1}{2} (\boldsymbol{f}_a + \boldsymbol{f}_b) \right] \otimes \underbrace{\sum_{k \in \Lambda_{ab}^k} \boldsymbol{N}_k A_k}_{\boldsymbol{C}_{ab}} \tag{3.29}$$

where Λ_{ab}^k is the set of facets belonging to edge ab, A_k is the area of a singular facets and N_k its outward normal. The two expressions, (3.27) and (3.29), must be equal and the desired relation can be identified

$$\boldsymbol{C}_{ab} = 2V_a V_b \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a) \tag{3.30}$$

3.3.1 Total Lagrangian SPH discrete formulation

The above weak statement (3.16) is spatially discretized [20]. In particular, the external virtual work $\delta\mathcal{W}_{ext}$ becomes

$$\begin{aligned} \delta\mathcal{W}_{ext} &\approx \sum_B A_B \delta\boldsymbol{v}(\boldsymbol{X}_B) \cdot \boldsymbol{t}(\boldsymbol{X}_B, t) + \sum_a V_a \delta\boldsymbol{v}(\boldsymbol{X}_a) \cdot \boldsymbol{b}_a \\ &\approx \sum_a V_a \delta\boldsymbol{v}_a \cdot \left[\frac{A_a}{V_a}\boldsymbol{t}_a + \boldsymbol{b}_a\right] \end{aligned} \tag{3.31}$$

where B represent the set of particles in the boundaries of the domain, $\boldsymbol{t}$ is the traction vector acting on the area A (remember that $A_a = 0$ when $a \notin B$) and $\boldsymbol{b}$ is the body force. Additionally, the internal virtual work contribution $\delta\mathcal{W}_{int}$ is written as

$$\begin{aligned} \delta\mathcal{W}_{int} &\approx \sum_b V_b \boldsymbol{P}_b : \nabla_0 \delta\boldsymbol{v}(\boldsymbol{X}_b) \\ &\approx \sum_b V_b \boldsymbol{P}_b : \left(\sum_{a \in \Lambda_b} V_a \ \delta\boldsymbol{v}_a \otimes \tilde{\nabla}_0 \tilde{W}_a(\boldsymbol{X}_b) \right) \\ &= \sum_a V_a \delta\boldsymbol{v}_a \cdot \left(\sum_{b \in \Lambda_a} \underbrace{V_b \boldsymbol{P}_b \cdot \tilde{\nabla}_0 \tilde{W}_a(\boldsymbol{X}_b)}_{\boldsymbol{T}_{ab}} \right) \end{aligned} \tag{3.32}$$

combining expression (3.31) and (3.32) and integrating using the cloud of particles as quadrature points, the resulting discrete weak formulation is

$$\begin{aligned} \rho \frac{\partial \boldsymbol{v}_a}{\partial t} &= -\sum_{b \in \Lambda_a} \boldsymbol{T}_{ab} + \frac{A_a}{V_a}\boldsymbol{t}_a + \boldsymbol{b}_a \\ \frac{\partial J_a}{\partial t} &= \boldsymbol{H}_a : \sum_{b \in \Lambda_a} V_b \ \boldsymbol{v}_b \otimes \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a) \end{aligned} \tag{3.33}$$

where the discrete co-factor $\boldsymbol{H}_a = J\boldsymbol{F}_a^{-T}$ (3.2) is evaluated by writing the deformation gradient $\boldsymbol{F}_a$ as

$$\boldsymbol{F}_a = \sum_{b \in \Lambda_a} V_b \ \boldsymbol{x}_b \otimes \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a) \tag{3.34}$$

with $\boldsymbol{x}_b$ position of particle b at the current configuration. Another useful representation of the deformation gradient is written in terms of $\boldsymbol{v}$, as is shown later in this work.

3.3.2 Riemann-based Upwinding stabilization

As mentioned before, the integration is done using the particles themselves as quadrature points. This operation may lead to instabilities such as zero-energy modes and pressure instability, consequently the scheme could encounter a break-down due to error accumulation.

The numerical stabilization for zero-energy modes and pressure instability is introduced

via a consistent characteristic-based Riemann solver [19, 21, 27], adopted also in finite volumes method [14, 15].
In Lagrangian dynamics, it is possible for two distinct domains (i.e. V_a and V_b) to come into contact with each other after some time t. These impacts create two types of shock waves (figure 3.4), p-waves (pressure waves moving with speed c_p through the material) and s-waves (transverse waves moving with speed c_s)[5], that could propagate between the particle domains.

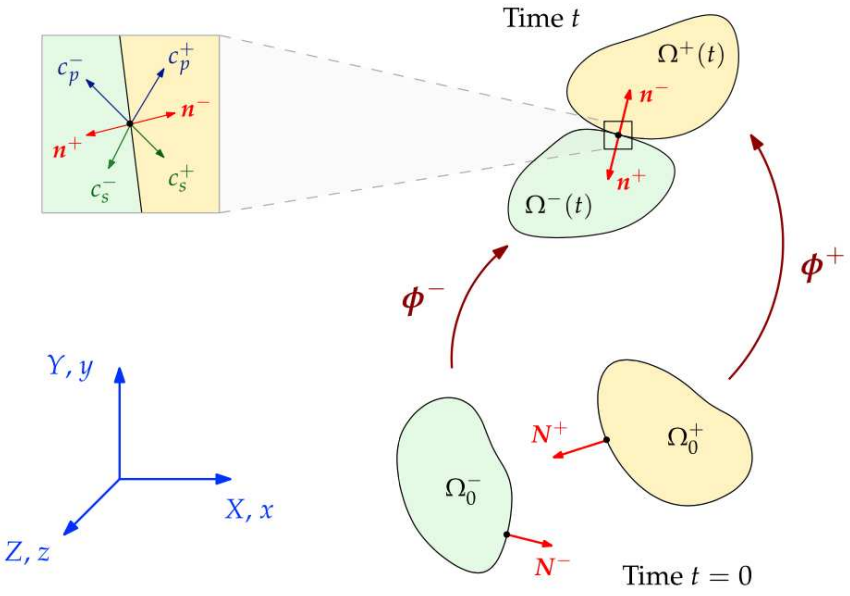


Figure 3.4: Contact mechanics [14]

To model these interactions and maintain numerical stability, the flux is evaluated using a Riemann solver based on the Rankine–Hugoniot jump conditions (3.4) and (3.7). The flux is computed across a discontinuity surface defined by the outward normal vector $\boldsymbol{N}_{ab}$, for each couple of neighbor particles.
In practice, an approximate local Riemann problem is solved at each particle interface. The goal is to improve the quality of the solution and reduce numerical oscillations caused by sharp jumps in the state variables. This is achieved by replacing the unstable flux in (3.8) with

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ab}} = \frac{1}{2}\left[\boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ab}}(\boldsymbol{\mathcal{U}}_f^-) + \boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ab}}(\boldsymbol{\mathcal{U}}_f^+)\right] - \frac{1}{2}|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ab}}|(\boldsymbol{\mathcal{U}}_f^+ - \boldsymbol{\mathcal{U}}_f^-) \tag{3.35}$$

The first term is a simple average from left and right state while the second term can be seen as a numerical flux that dumps the instabilities, it is proportional to the flux jump [6] and $\boldsymbol{\mathcal{A}}$ is the flux Jacobian computed as $\partial\boldsymbol{\mathcal{F}}/\partial\boldsymbol{\mathcal{U}}$.
These numerical fluxes can be decomposed into the summation of an average state and an upwinding numerical stabilization term. Indeed,

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ab}} = -\underbrace{\begin{bmatrix} \boldsymbol{P}^{Ave}\boldsymbol{N}_{ab} \\ \boldsymbol{H}_a : (\boldsymbol{v}^{Ave} \otimes \boldsymbol{N}_{ab}) \end{bmatrix}}_{\text{Unstable flux}} - \underbrace{\begin{bmatrix} \boldsymbol{P}^{Stab}\boldsymbol{N}_{ab} \\ \boldsymbol{H}_a : (\boldsymbol{v}^{Stab} \otimes \boldsymbol{N}_{ab}) \end{bmatrix}}_{\text{Upwinding Stabilization}} \tag{3.36}$$

[5] In this work only volumetric effects are considered, then s-waves will not be considered.
[6] This ensures that the stabilization is applied only where necessary, when discontinuities occur. In smooth regions, where the jump is negligible the solution does not need to be smoothed.

where the expression for the stabilised first Piola–Kirchhoff stresses and velocity are

$$\boldsymbol{P}^{Stab} = \left[\boldsymbol{S}^{v}_{ab} \cdot (\boldsymbol{v}^{+}_{f} - \boldsymbol{v}^{-}_{f})\right] \otimes \boldsymbol{N}_{ab}; \qquad \boldsymbol{v}^{Stab} = \boldsymbol{S}^{P}_{ab}\left[(\boldsymbol{P}^{+}_{f} - \boldsymbol{P}^{-}_{f})\boldsymbol{N}_{ab}\right]; \tag{3.37}$$

and the stabilization matrices $\boldsymbol{S}^{P}_{ab}$ and $\boldsymbol{S}^{v}_{ab}$ are defined as in [14]

$$\boldsymbol{S}^{v}_{ab} = \frac{1}{2}c_p(\boldsymbol{n}_{ab} \otimes \boldsymbol{n}_{ab}); \qquad \boldsymbol{S}^{P}_{ab} = \frac{1}{2c_p}(\boldsymbol{n}_{ab} \otimes \boldsymbol{n}_{ab}) \tag{3.38}$$

In equation (3.38) the outward unit normal is defined using the current positions as $\boldsymbol{n}_{ab} = (\boldsymbol{x}_b - \boldsymbol{x}_a)/ \parallel \boldsymbol{x}_b - \boldsymbol{x}_a \parallel$ while the symbols $\{\boldsymbol{v}^{+,-}_{ab}, \boldsymbol{P}^{+,-}_{ab}\}$ in expression (3.37) represent the left and right states of $\{\boldsymbol{v}, \boldsymbol{P}\}$ at fictitious interface f. The standard procedure is to consider the values of problem variables at particle position (i.e. $\boldsymbol{v}_a$, $\boldsymbol{v}_b$, $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$).

3.3.3 Complete Upwind-SPH formulation

It is now possible to include Riemann-based stabilization terms in the total Lagrangian SPH discrete formulation.
To achieve this, (3.36) is inserted in (3.33) substituting $\{\boldsymbol{P}^{Ave}, \boldsymbol{v}^{Ave}\}$ and obtaining, after some manipulation, the final discrete formulation

$$\begin{aligned} \rho\frac{\partial \boldsymbol{v}_a}{\partial t} &= -\sum_{b\in\Lambda_a} \boldsymbol{T}_{ab} + \frac{A_a}{V_a}\boldsymbol{t}_a + \boldsymbol{b}_a + \sum_{b\in\Lambda_a} \boldsymbol{\mathcal{D}}^{ab}_{\boldsymbol{v}} \\ \frac{\partial J_a}{\partial t} &= \boldsymbol{H}_a : \sum_{b\in\Lambda_a} V_b\ \boldsymbol{v}_b \otimes \tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a) + \sum_{b\in\Lambda_a} \mathcal{D}^{ab}_{p} \end{aligned} \tag{3.39}$$

where the dissipative terms are defined as

$$\boldsymbol{\mathcal{D}}^{ab}_{\boldsymbol{v}} = \tau_{\boldsymbol{v}} \sum_{b\in\Lambda_a} 2V_b\boldsymbol{P}^{Stab} \cdot \tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a) \qquad \mathcal{D}^{ab}_{p} = \tau_p \sum_{b\in\Lambda_a} 2V_b\frac{\boldsymbol{v}^{Stab}}{\rho} \cdot \left[H_a\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)\right] \tag{3.40}$$

where $\tau_{\boldsymbol{v}}$ and τ_p are parameters that controls the magnitude of the dissipation. To obtain (3.40) recall remark 5.

Remark 6. In this case, due to the corrections applied, $\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a) \neq -\tilde{\nabla}_0\tilde{W}_a(\boldsymbol{X}_b)$. Consequently the dissipation term $\boldsymbol{\mathcal{D}}^{ab}_{\boldsymbol{v}}$ will not satisfy the global momentum conservation requirement. However, it is possible to fulfill the condition by replacing $\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)$ with his average counterpart defined as

$$\tilde{\nabla}_0\tilde{W}^{Ave}_b(\boldsymbol{X}_a) = \frac{1}{2}\left(\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a) - \tilde{\nabla}_0\tilde{W}_a(\boldsymbol{X}_b)\right) \tag{3.41}$$

Same procedure can be done in $\mathcal{D}^{ab}_{p}$ resulting in

$$\left[\boldsymbol{H}_a\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)\right]^{Ave} = \frac{1}{2}\left[\boldsymbol{H}_b\tilde{\nabla}_0\tilde{W}_a(\boldsymbol{X}_b) - \boldsymbol{H}_a\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)\right] \tag{3.42}$$

3.3.4 Numerical entropy production

In this section is shown a procedure to verify the consistency and local conservation of the stabilization terms by using a semi-discrete version of the Coleman-Noll procedure as in [22]. Recall (3.19), its discrete version reads as

$$\sum_a V_a\frac{d\mathcal{H}}{dt} = \sum_a V_a\left[\boldsymbol{v}_a \cdot \rho\frac{\partial \boldsymbol{v}_a}{\partial t} - p_a\frac{\partial J_a}{\partial t}\right] \tag{3.43}$$

Substituting (3.39) into (3.43) yields

$$\sum_a V_a \frac{d\mathcal{H}}{dt} = \sum_a V_a \sum_{b\in\Lambda_a^b} (V_b \boldsymbol{P}_b \cdot \nabla_0 W_a(\boldsymbol{X}_b)) \cdot \boldsymbol{v}_a + \dot{\Pi}_{ext} + \underbrace{\sum_a \sum_{b\in\Lambda_a^b} \left(\boldsymbol{v}_a \cdot \boldsymbol{\mathcal{D}_v}^{ab} - p_a \mathcal{D}_p^{ab}\right)}_{-\mathcal{D}_{total}} \tag{3.44}$$

where $\dot{\Pi}_{ext}$ denotes the power introduced by external forces expressed as

$$\dot{\Pi}_{ext} = \sum_a (V_a \boldsymbol{v}_a \cdot \boldsymbol{b}_0 + A_0^a \boldsymbol{v}_B^a \cdot \boldsymbol{t}_B^a) \tag{3.45}$$

Since $\nabla_0 W_b(\boldsymbol{X}_a) = -\nabla_0 W_a(\boldsymbol{X}_b)$, the first term of equation (3.44) vanishes due to anti symmetry and the expression simplifies to

$$\sum_a V_a \frac{d\mathcal{H}}{dt} - \dot{\Pi}_{ext} = -\mathcal{D}_{total} \tag{3.46}$$

It is now sufficient to demonstrate that the right-hand side of equation (3.46) is always non positive, or equivalently that $\mathcal{D}_{total} \geq 0$, to show that the stabilization terms introduced fulfill the second law of thermodynamics. The term $\mathcal{D}_{total}$ can be written as

$$\begin{aligned} \mathcal{D}_{total} &= -\left(\sum_a \sum_{b\in\Lambda_a} \boldsymbol{v}_a \boldsymbol{\mathcal{D}_v}^{ab} - p_a \mathcal{D}_p^{ab}\right) \\ &= -\left(\sum_a \sum_{b\in\Lambda_a} \boldsymbol{v}_b \boldsymbol{\mathcal{D}_v}^{ba} - p_b \mathcal{D}_p^{ba}\right) \end{aligned} \tag{3.47}$$

By summing the first and second line of equation (3.47) taking into account the anti symmetry of the stabilization terms, an equivalent expression for $\mathcal{D}_{total}$ is

$$\mathcal{D}_{total} = \frac{1}{2}\left(\sum_a \sum_{b\in\Lambda_a} (\boldsymbol{v}_b - \boldsymbol{v}_a) \boldsymbol{\mathcal{D}_v}^{ab} - (p_b - p_a)\mathcal{D}_p^{ab}\right) \tag{3.48}$$

Recall the complete expressions for the stabilization terms

$$\begin{aligned} \boldsymbol{\mathcal{D}_v}^{ab} &= 2V_b \left[\boldsymbol{S}_{ab}^v \cdot (\boldsymbol{v}_b - \boldsymbol{v}_a)\right] \otimes \boldsymbol{N}_{ab} \cdot \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a)^{Ave} \\ \boldsymbol{\mathcal{D}}_p^{ab} &= -2V_b \frac{1}{\rho} \boldsymbol{S}_{ab}^p \left[(p_b - p_a)\boldsymbol{H}^{Ave} \boldsymbol{N}_{ab}\right] \cdot \left[H_a \tilde{\nabla}_0 \tilde{W}_b(\boldsymbol{X}_a)\right]^{Ave} \end{aligned} \tag{3.49}$$

where the two matrices $\boldsymbol{\mathcal{S}^v}_{ab}$ and $\boldsymbol{\mathcal{S}}_{ab}^p$ are symmetric and positive semi-definite meaning that for any vector $\boldsymbol{v}$, it holds that $\boldsymbol{v}^T \boldsymbol{\mathcal{S}}_{ab} \boldsymbol{v} \geq 0$.. The terms $\boldsymbol{v}_b - \boldsymbol{v}_a$ and $p_b - p_a$ have sign but, multiplied by the terms present in (3.48) and the positive semi-definite matrices, their final contribution is positive. Moreover, by construction, the dot product $\boldsymbol{N}_{ab} \cdot \nabla_0 W_b(\boldsymbol{X}_a)$ is also positive. Recall that the minus sign present in front of the second equation comes from the constitutive law $\boldsymbol{P} = -p\boldsymbol{H}$. The remaining quantities are constant positive parameters[7]. Given these considerations, it is proved that the dissipation term $\mathcal{D}_{total}$ is positive.

[7]The term $\boldsymbol{H}^{Ave}$ is computed as 0.5 $(\boldsymbol{H}_a + \boldsymbol{H}_b)$, then multiplied by the other $\boldsymbol{H}^{Ave}$ present in the formula the total contribute is a positive quantity.

3.3.5 Linear reconstruction procedure

The stabilization terms act as diffusive terms, introducing an artificial viscosity aimed at reducing spurious oscillations or discontinuities caused by shock waves. However, this results in energy dissipation within the system, making it crucial to minimize such losses. As previously said, the amount of dissipation is proportional to the jump of the problem variables at particle interfaces. Instead of assuming the problem variables to be constant within the support, their values are locally approximated at the interface using a linear reconstruction procedure using the set of neighboring particles[8].
The reconstruction procedure [19], used to evaluate the problem variables at the interface f between neighboring particles $\{\boldsymbol{v}^{+,-}, \boldsymbol{P}^{+,-}\}$, is now described.
The procedure consists in reconstructing the velocity at the mid-edge f connecting particles a and b. At the same time, the stresses on each particle (that is, $\boldsymbol{P}(\boldsymbol{F}_a)$ and $\boldsymbol{P}(\boldsymbol{F}_b)$) are considered and then reconstructed at the interface f , namely $\{\boldsymbol{P}_f^-, \boldsymbol{P}_f^+\}$. For any individual component of $\boldsymbol{\mathcal{U}}$, the linear reconstructed value at any position $\boldsymbol{X}$ is derived as

$$\boldsymbol{\mathcal{U}}(\boldsymbol{X}) = \boldsymbol{\mathcal{U}}_a + \boldsymbol{\mathcal{G}}_a \cdot (\boldsymbol{X} - \boldsymbol{X}_a) \tag{3.50}$$

A least square procedure is adopted to compute an appropriate particle gradient operator $\boldsymbol{\mathcal{G}}_a$. To do this an objective functional Π is introduced

$$\Pi(\boldsymbol{\mathcal{G}}_a) = \frac{1}{2} \sum_{b \in \Lambda_a} \frac{1}{\| \boldsymbol{d}_{ab} \|^2} \left[\boldsymbol{\mathcal{U}}_b - (\boldsymbol{\mathcal{U}}_a + \boldsymbol{\mathcal{G}}_a \cdot \boldsymbol{d}_{ab})\right]^2 \tag{3.51}$$

Here, $b \in \Lambda_a$ represents the set of neighboring particles, $\boldsymbol{d}_{ab}$ the distance vector between a and b. Taking directional derivative of (3.51) with respect to $\boldsymbol{\mathcal{G}}_a$, yields

$$0 = D\Pi(\boldsymbol{\mathcal{G}}_a) = \sum_{b \in \Lambda_a} \frac{1}{\| \boldsymbol{d}_{ab} \|^2} \left[\boldsymbol{\mathcal{U}}_b - (\boldsymbol{\mathcal{U}}_a + \boldsymbol{\mathcal{G}}_a \cdot \boldsymbol{d}_{ab})\right] \boldsymbol{d}_{ab} \tag{3.52}$$

Rearranging[9], the final expression for $\boldsymbol{\mathcal{G}}_a$ is

$$\boldsymbol{\mathcal{G}}_a = \left[\sum_{b \in \Lambda_a} \boldsymbol{N}_{ab} \otimes \boldsymbol{N}_{ab}\right]^{-1} \cdot \sum_{b \in \Lambda_a} \left(\frac{\boldsymbol{\mathcal{U}}_b - \boldsymbol{\mathcal{U}}_a}{\| \boldsymbol{d}_{ab} \|}\right) \boldsymbol{N}_{ab} \tag{3.53}$$

with $\boldsymbol{N}_{ab}$ as the unit vector in the direction of $\boldsymbol{d}_{ab}$.

3.3.6 Shock wave propagation speed c_p

As discussed in Section 3.3.2, when the stabilization based on the Riemann solver is considered, two wave speeds become relevant: the longitudinal shock propagation speed c_p and the transverse one c_s. However, since the material model considered in this work (see equation (3.12)) neglects the shear resistance, only c_p is used for stabilization purposes. Initially, the wave speed c_p can be assumed to be constant in time, and in a linear elastic material, equal to its reference value

$$c_p^0 = \sqrt{\frac{\kappa}{\rho}} \tag{3.54}$$

However, under large deformations, the configuration of the material, and so the volume assigned to each particle, can change shape significantly. For example, if the volume

[8] A linear function is exactly reconstructed in this way, resulting in no jump at the interface and no dissipation for a linear field. Moreover, if a linear function is reproduced exactly, then the reconstruction procedure is correctly implemented.

[9] To rearrange it has been used the vectors property $(\boldsymbol{a} \otimes \boldsymbol{b}) \cdot \boldsymbol{x} = (\boldsymbol{a} \cdot \boldsymbol{x})\ \boldsymbol{b} \quad \forall\ \boldsymbol{x}$.

becomes highly stretched, the wave propagation in one direction should be higher than in another. To avoid numerical instabilities, especially when the material is stretched, it is important to update the propagation speed c_p to project the current deformation effects in the reference configuration. Following the approach proposed in [17], the shock wave speed should be adjusted based on the current configuration. To quantify the deformation of a particle, we consider the right Cauchy-Green strain tensor

$$\boldsymbol{C}_a = \boldsymbol{F}_a^T \boldsymbol{F}_a \tag{3.55}$$

Its eigenvalues λ_i are obtained by solving

$$\det(\boldsymbol{C}_a - \lambda_i \mathbb{I}) = 0 \tag{3.56}$$

these eigenvalues represent the squares of the principal stretches and thus describe how the material is deformed relatively to its reference configuration. To ensure stability, the updated value of c_p for a particle can be defined as

$$c_p = \frac{c_p^0}{\sqrt{\lambda_{min}}} \quad \forall \text{ particle} \tag{3.57}$$

where λ_{min} is the smallest eigenvalue of $\boldsymbol{C}_a$. This approach increases c_p, and so the magnitude of stabilization, in regions where the material is highly deformed in one direction, improving the numerical stability locally. However, in case of extreme compression, the corresponding minimum eigenvalue of the right Cauchy-Green tensor can become very small, potentially leading to numerical instability causing the code to break down. In general, if deformation is limited, for example when the continuum exhibits oscillatory elastic behavior, assuming a constant c_p may be sufficient.

Chapter 4

Time integration

In Smoothed Particle Hydrodynamics time integration schemes plays an important role in maintaining accuracy and stability over time. Traditionally, explicit time integration schemes have been the standard approach in the literature due to their simplicity and ease of implementation. However, these methods suffer from a severe restriction on the time step, given by the Courant–Friedrichs–Lewy (CFL) condition. This restriction can significantly limit computational efficiency, especially in simulations involving high-resolution domains or large deformations.
To overcome this limitation, implicit time integration could be an alternative.
This chapter first presents an explicit time integration scheme commonly used in the SPH literature and then introduces an implicit scheme based on the Crank–Nicolson method. The proposed implicit scheme will be tested and validated in the following chapters.

4.1 Explicit time discretization

An explicit one step two-stage Total Variation Diminishing Runge-Kutta (TVD-RK) scheme has been adopted to solve the evolutional problem [15, 19, 20]. The following equations describes the update of the problem variables from time step t_n to t_{n+1}.

$$\begin{aligned} \boldsymbol{\mathcal{U}}_a^\star &= \boldsymbol{\mathcal{U}}_a^n + \Delta t\, \dot{\boldsymbol{\mathcal{U}}}_a^n \left(\boldsymbol{\mathcal{U}}_a^n, t^n\right) \\ \boldsymbol{\mathcal{U}}_a^{\star\star} &= \boldsymbol{\mathcal{U}}_a^\star + \Delta t\, \dot{\boldsymbol{\mathcal{U}}}_a^\star \left(\boldsymbol{\mathcal{U}}_a^\star, t^{n+1}\right) \\ \boldsymbol{\mathcal{U}}_a^{n+1} &= \frac{1}{2}\left(\boldsymbol{\mathcal{U}}_a^n + \boldsymbol{\mathcal{U}}_a^{\star\star}\right) \end{aligned} \tag{4.1}$$

The geometry (position of each particle) is also updated using the above explicit scheme. The problem variables $\boldsymbol{\mathcal{U}} = \{\boldsymbol{v}_a, J_a, \boldsymbol{x}_a\}$[1] with $a = 1, \ldots, N$ are updated using (4.1) resulting in a monolithic time integration procedure.
The maximum time step $\Delta t = t_{n+1} - t_n$ is controlled by the Courant-Friedrichs-Lewy (CFL) condition [7], which ensures that the propagation of shocks remains physically consistent with the discrete resolution of the model. In particular, the time step must be small enough to allow information to travel across the characteristic length scale within a single time increment. The CFL number is defined as:

$$\alpha_{CFL} = \frac{\Delta t\; c_{p,max}}{d_{min}} \tag{4.2}$$

[1] Also $\boldsymbol{F}$ is updated within (4.1) in order to avoid to re-compute the kernels and kernel gradients for each particle. To do so the deformation gradient is written as $\boldsymbol{F}^{step+1} = \boldsymbol{F}^{step} + \dot{\boldsymbol{F}}$, notice that $\dot{\boldsymbol{F}}$ depends on $\boldsymbol{v}$.

This condition is satisfied when $\alpha_{CFL} \leq 1$. Solving for the time step yields

$$\Delta t = \alpha_{CFL} \frac{d_{min}}{c_{p,max}} \tag{4.3}$$

where $c_{p,max}$ is the the maximum p-wave speed, d_{min} is the minimum particle spacing within the domain and α_{CFL} acts as a stability coefficient that has to be ≤ 1. For instance, in [22], α_{CFL} is kept equal to 0.3, this is a conservative choice to ensure both accuracy and stability.
From equation (4.3), it is clear that the time step is inversely proportional to both the discretization refinement and the bulk modulus κ of the material (see equation 3.54). The algorithm 1 sums up the description of the explicit Upwind SPH methodology.

Algorithm 1 SPH explicit time integration

1: **Input**: $\boldsymbol{\mathcal{U}}_a^n$ where $\boldsymbol{\mathcal{U}} = [\boldsymbol{v}, J]^T$
2: Assign $\boldsymbol{\mathcal{U}}^{old} = \boldsymbol{\mathcal{U}}^n$ and $\boldsymbol{x}^{old} = \boldsymbol{x}^n$
3: Evaluate *p-wave* speed c_p
4: Compute time increment Δt (see equation (4.3))
5: **for** TVD-RK time integrator = 1 to 2 **do**
6: Compute RHS of mixed-based system: $\dot{\boldsymbol{v}}$ and $\dot{J}$ (see equation (3.39))
7: Evolve $\boldsymbol{\mathcal{U}}_a$, $\boldsymbol{x}_a$
8: Impose boundary conditions
9: **end for**
10: Update $\boldsymbol{\mathcal{U}}_a^{n+1}$, $\boldsymbol{x}_a^{n+1}$
11: **Output**: $\boldsymbol{\mathcal{U}}_a^{n+1}$, $\boldsymbol{x}_a^{n+1}$

4.2 Implicit time discretization

Unlike the method described above, an implicit time integration scheme calculates the set of problem variables at time step $n+1$ based on the variables at the same time step. This can be expressed as

$$\boldsymbol{\mathcal{U}}_a^{n+1} = \boldsymbol{\mathcal{U}}_a^n + \Delta t\, \dot{\boldsymbol{\mathcal{U}}}_a^{n+1}(\boldsymbol{\mathcal{U}}_a^{n+1}, t^{n+1}) \tag{4.4}$$

The expression (4.4) represents the backward Euler scheme, which is an implicit first-order method for time integration. While it is unconditionally stable, it is also numerically dissipative and only first-order accurate, meaning that the global truncation error is proportional to $\mathcal{O}(\Delta t)$. To improve accuracy, the Crank-Nicolson scheme is introduced by averaging the explicit and implicit Euler schemes

$$\boldsymbol{\mathcal{U}}_a^{n+1} = \boldsymbol{\mathcal{U}}_a^n + \Delta t\, \frac{1}{2} \left(\dot{\boldsymbol{\mathcal{U}}}_a^{n+1}(\boldsymbol{\mathcal{U}}_a^{n+1}, t^{n+1}) + \dot{\boldsymbol{\mathcal{U}}}_a^{n}(\boldsymbol{\mathcal{U}}_a^{n}, t^{n}) \right) \tag{4.5}$$

Crank-Nicolson is an implicit scheme that achieves second-order accuracy $\mathcal{O}(\Delta t^2)$. This formulation is stable and more accurate, reducing numerical dissipation. The system of equations to be solved, (3.39), depends on both velocity $\boldsymbol{v}$ and position $\boldsymbol{x}$. Since the Crank–Nicolson scheme handles only first-order time derivatives, an additional equation is introduced to relate velocity and position, $\dot{\boldsymbol{x}} = \boldsymbol{v}$.
An alternative approach would be to use a different time integration scheme, such as the Newmark method, which is applied in finite elements, as shown in [25, 28].
Given the nonlinearity of this formulation, the two governing equations result in a non-linear system of the form $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}) = 0$, where $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}})$ represents the residual vector and $\boldsymbol{\mathcal{U}} = \{\boldsymbol{v}, J, \boldsymbol{x}\}$. To solve this non-linear system, a Newton-Raphson iterative scheme is adopted, which involves linearizing the residual equations around an initial guess. At each

iteration, the linearized system is solved to compute corrections to the current guess until convergence is achieved. The linearized system is written as

$$\begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial \boldsymbol{x}} \\ \frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial \boldsymbol{x}} \\ \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial \boldsymbol{x}} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{v} \\ \Delta J \\ \Delta \boldsymbol{x} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\mathcal{R}}_{\boldsymbol{v}} \\ \boldsymbol{\mathcal{R}}_{J} \\ \boldsymbol{\mathcal{R}}_{\boldsymbol{x}} \end{bmatrix} \tag{4.6}$$

where $\boldsymbol{\mathcal{R}}_{\boldsymbol{v}}$ and $\boldsymbol{\mathcal{R}}_{J}$ are the residual associated with the problem variables ($\boldsymbol{v}$ and J), the Jacobian matrix is a block matrix and contains the partial derivatives of the residuals with respect to $\boldsymbol{v}$ and J. Finally, $\Delta \boldsymbol{v}$ and ΔJ represent corrections to the current guess.
For completeness, the residuals are defined as

$$\begin{aligned} \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}(\boldsymbol{v}_a, J_a) &= \rho\ \boldsymbol{v}_a^{n+1,k} - \rho\ \boldsymbol{v}_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{\boldsymbol{v}}_a^{n+1,k} + \dot{\boldsymbol{v}}_a^{n}\right) = 0 \\ \boldsymbol{\mathcal{R}}_{J}(\boldsymbol{v}_a, J_a) &= J_a^{n+1,k} - J_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{J}_a^{n+1,k} + \dot{J}_a^{n}\right) = 0 \\ \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(\boldsymbol{v}_a, J_a) &= \boldsymbol{x}_a^{n+1,k} - \boldsymbol{x}_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{\boldsymbol{x}}_a^{n+1,k} + \dot{\boldsymbol{x}}_a^{n}\right) = 0 \end{aligned} \tag{4.7}$$

where the time derivatives of the problem variables (i.e. $\dot{v}$ and $\dot{J}$) are the right hand side of (3.39).
Since their values at time step $n+1$ are not known a priori, the solution process involves the following steps:

1. *Initialization:* Start with an initial guess for the problem variables $\boldsymbol{\mathcal{U}}^{n+1,0}$.
   A common and intelligent choice is to use the values from the previous time step $\boldsymbol{\mathcal{U}}^{n}$ as initial guess.

2. *Linearization:* At each iteration k, the residuals are evaluated using the current guess $\boldsymbol{\mathcal{U}}^{n+1,k}$. The Jacobian is computed and the corrections $\Delta \boldsymbol{v}$ and ΔJ are determined by solving the following linear system

$$\boldsymbol{J}(\boldsymbol{\mathcal{U}}^{n+1,k})\Delta \boldsymbol{\mathcal{U}} = -\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1,k}) \tag{4.8}$$

3. *Update:* Update the guess using the computed correction:

$$\boldsymbol{\mathcal{U}}^{n+1,k+1} = \boldsymbol{\mathcal{U}}^{n+1,k} + \Delta \boldsymbol{\mathcal{U}} \tag{4.9}$$

4. *Convergence check:* Check if the residual norm $\parallel \boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1,k+1}) \parallel$ is below a predefined tolerance.[2] If the condition is satisfied, the process stops. Otherwise, repeat from step 2.

The iterative process continues until the solution converges to a consistent set of variables at time step $n+1$.
The novel procedure is summarized in algorithm 2

[2]The check in the residual norm is preferably done with the relative residual obtained dividing $\parallel \boldsymbol{R}^{n+1,k} \parallel$ by $\parallel \boldsymbol{R}^{n+1,0} \parallel$, to avoid numerical problems when dealing with small residual norms. Moreover, the residuals components computed in the boundary particles are not considered.

Algorithm 2 SPH implicit time integration, Newton-Raphson iteration

Input: $\boldsymbol{\mathcal{U}}_a^n$ where $\boldsymbol{\mathcal{U}} = [\boldsymbol{x},\ \boldsymbol{v},\ J]^T$, Δt
Assign $\boldsymbol{\mathcal{U}}^{old} = \boldsymbol{\mathcal{U}}^n$
Compute residual of mixed-based system: $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1,k+1})$
for $k \leq it_{max}$ and $\| \boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1,k+1}) \| \geq tol$ **do**
    Compute the Jacobian: $\boldsymbol{J}(\boldsymbol{\mathcal{U}}^{n+1,k})$
    Solve the linear system and compute $\Delta\boldsymbol{\mathcal{U}}$ (see equation (4.6))
    Impose boundary conditions
    Evolve $\boldsymbol{\mathcal{U}}^{n+1,k+1}$
    Evolve deformation gradient $\boldsymbol{F}^{n+1,k+1}$
    Compute residual of mixed-based system: $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1,k+1})$
end for
Output: $\boldsymbol{\mathcal{U}}_a^{n+1}$

Chapter 5

Numerical examples: 1D tension problem

In this chapter, a one-dimensional (1D) tension problem is solved using a total Lagrangian SPH formulation and a linear elastic constitutive law. This simplified test case allows for a straightforward assessment of the proposed 1D algorithm, ensuring its accuracy and stability before extending it to more complex scenarios. Firstly, the problem is solved using the explicit Runge-Kutta time integration scheme, following the work done in [8] and [4][1].
Then, the implicit time integration scheme proposed in Chapter 4 is adopted and compared with explicit SPH and explicit FEM.

5.1 1D Total Lagrangian SPH formulation

First of all, working on a 1D problem allows for some simplifications. By definition, the volume changes can be directly characterized through the deformation gradient F, which is equivalent to its Jacobian J, therefore

$$H = JF^{-1} = 1; \qquad P = J\sigma F^{-1} = \sigma \tag{5.1}$$

where P is the first Piola-Kirchhoff stress tensor, and σ is the Cauchy stress tensor.
The conservation equations for linear momentum conservation and volume conservation in the total Lagrangian formulation reduces to the following 1D formulation

$$\begin{aligned} \rho_0 \frac{\partial v}{\partial t} - \nabla_0 \, \sigma - b = 0 \\ \frac{\partial J}{\partial t} - \nabla_0 v = 0 \end{aligned} \tag{5.2}$$

or in a more compact way as

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}}{\partial X} = \mathcal{S} \tag{5.3}$$

[1]This problem was used to show tension instability by J. Bonet and S. Kulasegaram in [4] and can be solved using SPH approach only with a total Lagrangian approach and using corrected SPH. Indeed, by using linear kernel function with a limited support can be easily shown that with a standard Eulerian approach the linear system matrix has negative eigenvalues leading to instabilities and unstable solutions.

where $\mathcal{U}$ represent the problem variables and $\mathcal{F}$ the flux vector. Following the approach shown in Chapter 3, the discrete 1D version of equation (3.33) reads as

$$\begin{aligned}\rho\frac{\partial v_a}{\partial t} &= -\sum_{b\in\Lambda_a} V_b\ \sigma_b\ \tilde{\nabla}_0\tilde{W}_a(X_b) + b_a \\ \frac{\partial J_a}{\partial t} &= \sum_{b\in\Lambda_a} V_b\ v_b\ \tilde{\nabla}_0\tilde{W}_b(X_a)\end{aligned} \tag{5.4}$$

Finally, to close the system, the 1D linear elastic constitutive law[2] is applied

$$\sigma = -\kappa(J-1) \tag{5.5}$$

with κ being the bulk modulus[3].
Without proper stabilization, this formulation can suffer from numerical instabilities. To mitigate these effects, an upwind Riemann-based stabilization (described in Chapter 3) is introduced, adapting the approach used in FVM [12, 14]. The idea is to replace the flux at the interface between particles with a stabilized version

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ab}} = \frac{1}{2}\left[\boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ab}}(\boldsymbol{\mathcal{U}}^-_f) + \boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ab}}(\boldsymbol{\mathcal{U}}^+_f)\right] - \frac{1}{2}|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ab}}|(\boldsymbol{\mathcal{U}}^+_f - \boldsymbol{\mathcal{U}}^-_f) \tag{5.6}$$

This formulation includes a dissipative term proportional to the jump in problem variables at the interface.
The absolute value of the flux Jacobian matrix $|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ab}}| = |\frac{\partial \boldsymbol{\mathcal{F}}_N}{\partial \boldsymbol{\mathcal{U}}}|$ can be defined as

$$\begin{aligned}|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ab}}| &= \frac{1}{2}\sum_{\alpha=1}^{N_{eq}} |c_\alpha|\boldsymbol{\mathcal{R}}_\alpha\boldsymbol{\mathcal{L}}^T_\alpha \\ &= \frac{1}{2}c_\alpha\left(\boldsymbol{\mathcal{R}}_1\boldsymbol{\mathcal{L}}^T_1 + \boldsymbol{\mathcal{R}}_2\boldsymbol{\mathcal{L}}^T_2\right)\end{aligned} \tag{5.7}$$

where $\boldsymbol{\mathcal{R}}_\alpha$ and $\boldsymbol{\mathcal{L}}_\alpha$ are respectively the right and left eigenvectors of $|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ab}}|$ and c_α the wave speed. First of all, considering the constitutive equation (5.5), define the flux Jacobian matrix $\boldsymbol{\mathcal{A}}_N$ of the system of conservation laws obtained by taking the derivative of the system flux with respect to system variables v and J.

$$\begin{bmatrix} 0 & -\kappa/\rho \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -c_p^2 \\ 1 & 0 \end{bmatrix} \tag{5.8}$$

where c_p is the material wave speed. The eigenvalues of (5.8) are such that $\det(A-\lambda\mathrm{I}) = 0$, that gives $\lambda^2 - c_p^2 = 0$ or in matrix form

$$\Lambda = \begin{bmatrix} c_p & 0 \\ 0 & -c_p \end{bmatrix} \tag{5.9}$$

since both eigenvalues $\in \mathbb{R}$, the system is hyperbolic. To compute right and left eigenvectors, recall that the following conditions

$$c_\alpha\boldsymbol{\mathcal{R}}_\alpha = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{R}}_\alpha \qquad c_\alpha\boldsymbol{\mathcal{L}}^T_\alpha = \boldsymbol{\mathcal{L}}^T_\alpha\boldsymbol{\mathcal{A}} \tag{5.10}$$

Using the definition of right eigenvector, consider a vector $w = [v_1\ v_2]^T$, then

$$\begin{aligned}c_p v_1 &= -c_p^2 v_2 \\ c_p v_2 &= -v_1\end{aligned} \tag{5.11}$$

[2]To be able to couple the two equations, the constitutive law is written in terms of J.
[3]Bulk modulus and elastic constant are equal in 1D : $\kappa = E$

by substituting the second expression into the first one, and selecting $v_1 = 1$ for simplicity, the following right eigenvectors are obtained. Note that the same procedure when applied to compute the left eigenvectors gives

$$\boldsymbol{\mathcal{R}}_{1,2} = \begin{bmatrix} 1 \\ \pm\frac{1}{c_p} \end{bmatrix}; \qquad \boldsymbol{\mathcal{L}}_{1,2} = \begin{bmatrix} 1 \\ \pm\frac{1}{c_p} \end{bmatrix} \tag{5.12}$$

Finally, using (5.7) the final expression for the absolute value of the flux Jacobian matrix is

$$|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}}| = \begin{bmatrix} c_p & 0 \\ 0 & \frac{1}{c_p} \end{bmatrix} \tag{5.13}$$

Recalling remark 5, is possible to adapt these terms to SPH methods leading to

$$\begin{aligned} \sum_{b\in\Lambda_a} \mathcal{D}_v^{ab} &= \tau_{\boldsymbol{v}} \sum_{b\in\Lambda_a} V_b \cdot c_p(v_f^+ - v_f^-) N_{ab} \cdot \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \\ \sum_{b\in\Lambda_a} \mathcal{D}_p^{ab} &= \tau_p \sum_{b\in\Lambda_a} V_b \cdot \frac{1}{c_p} \cdot \frac{1}{\rho}(\sigma_f^+ - \sigma_f^-) N_{ab} \cdot \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \end{aligned} \tag{5.14}$$

The final 1D Upwind-SPH discrete formulation is written as

$$\begin{aligned} \rho\frac{\partial v_a}{\partial t} &= -\sum_{b\in\Lambda_a} V_b\ \sigma_b\ \tilde{\nabla}_0 \tilde{W}_a(X_b) + b_a + \sum_{b\in\Lambda_a} \mathcal{D}_v^{ab} \\ \frac{\partial J_a}{\partial t} &= \sum_{b\in\Lambda_a} V_b\ v_b\ \tilde{\nabla}_0 \tilde{W}_b(X_a) + \sum_{b\in\Lambda_a} \mathcal{D}_p^{ab} \end{aligned} \tag{5.15}$$

Time integration

In the literature, the explicit Total Variation Diminishing (TVD) Runge-Kutta scheme described in algorithm 1 is commonly employed.
However, this chapter aims to demonstrate the applicability of an implicit scheme. In particular, the implicit time integration scheme introduced in Chapter 4 is used to solve the system (5.15). As shown in algorithm 2, computing the time evolution of the problem variables requires evaluation of both the residuals and the Jacobian matrix within the Newton-Raphson method. Moreover a correct evaluation of these terms is also essential for the scheme's convergence.
The global system to solve can be written in the following compact form

$$\begin{bmatrix} \frac{\partial \mathcal{R}_v}{\partial v} & \frac{\partial \mathcal{R}_v}{\partial J} & \frac{\partial \mathcal{R}_v}{\partial x} \\ \frac{\partial \mathcal{R}_J}{\partial v} & \frac{\partial \mathcal{R}_J}{\partial J} & \frac{\partial \mathcal{R}_J}{\partial x} \\ \frac{\partial \mathcal{R}_x}{\partial v} & \frac{\partial \mathcal{R}_x}{\partial J} & \frac{\partial \mathcal{R}_x}{\partial x} \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta J \\ \Delta x \end{bmatrix} = - \begin{bmatrix} \mathcal{R}_v \\ \mathcal{R}_J \\ \mathcal{R}_x \end{bmatrix} \tag{5.16}$$

The computation of the residuals and their derivatives is detailed in this section. These derivatives have been obtained using Python's symbolic computation library SymPy [29]. Recall (4.7), then the complete expression for velocity residual $\mathcal{R}_{v,a}$ computed at particle a is

$$\begin{aligned} \mathcal{R}_{v,a} = \rho\ v_a^{n+1,k} - \rho\ v_a^n - \frac{1}{2}\ \Delta t \Bigg(& -\sum_{b\in\Lambda_a} V_b\ \left(\sigma_b^{n+1} + \sigma_b^n\right) \tilde{\nabla}_0 \tilde{W}_a(X_b) + (b_a^{n+1} + b_a^n) \\ & + \tau_v \sum_{b\in\Lambda_a} V_b \cdot c_p(v_f^+ - v_f^-) N_{ab} \cdot \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \Bigg) \end{aligned} \tag{5.17}$$

while the derivatives with respect to the problem variables v and J are

$$\begin{aligned}
\frac{\partial \mathcal{R}_{v,a}}{\partial v_a^{n+1}} &= 1 + \frac{1}{2}\,\Delta t \left(\tau_v \sum_{b\in\Lambda_a} V_b \cdot c_p\; N_{ab} \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \right) \\
\frac{\partial \mathcal{R}_{v,a}}{\partial v_b^{n+1}} &= -\frac{1}{2}\,\Delta t \left(\tau_v\; V_b \cdot c_p\; N_{ab} \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \right) \\
\frac{\partial \mathcal{R}_{v,a}}{\partial J_a^{n+1}} &= \frac{1}{2}\,\Delta t \left(V_a\; \kappa\; \tilde{\nabla}_0 \tilde{W}_a(X_a) \right) \\
\frac{\partial \mathcal{R}_{v,a}}{\partial J_b^{n+1}} &= \frac{1}{2}\,\Delta t \left(V_b\; \kappa\; \tilde{\nabla}_0 \tilde{W}_a(X_b) \right)
\end{aligned} \tag{5.18}$$

the Cauchy stress σ is defined by the pressure term $\kappa(J-1)$ and the reconstruction adopted in the stabilization terms considers the problem variables constant in the support.
The same procedure is shown for the Jacobian residual $\mathcal{R}_{J,a}$

$$\begin{aligned}
\mathcal{R}_{J,a} = J_a^{n+1,k} - J_a^n - \frac{1}{2}\,\Delta t \Bigg(& \sum_{b\in\Lambda_a} V_b\; \left(v_b^{n+1} + v_b^n \right) \tilde{\nabla}_0 \tilde{W}_a(X_b) \\
& + \tau_p \sum_{b\in\Lambda_a} V_b \cdot \frac{1}{c_p} \cdot \frac{1}{\rho}\; (\sigma_f^+ - \sigma_f^-) N_{ab} \cdot \tilde{\nabla}_0 \tilde{W}_b(X_a)^{Ave} \Bigg)
\end{aligned} \tag{5.19}$$

the derivatives are computed as

$$\begin{aligned}
\frac{\partial \mathcal{R}_{J,a}}{\partial v_a^{n+1}} &= -\frac{1}{2}\,\Delta t \left(V_a \tilde{\nabla}_0 \tilde{W}_a(X_a) \right) \\
\frac{\partial \mathcal{R}_{J,a}}{\partial v_b^{n+1}} &= -\frac{1}{2}\,\Delta t \left(V_b \tilde{\nabla}_0 \tilde{W}_b(X_a) \right) \\
\frac{\partial \mathcal{R}_{J,a}}{\partial J_a^{n+1}} &= 1 + \frac{1}{2}\,\Delta t \left(\tau_J \sum_{b\in\Lambda_a} V_b\; \frac{1}{c_p} \frac{\kappa}{\rho}\; N_{ab}\; \tilde{\nabla}_0 \tilde{W}_a(X_b) \right) \\
\frac{\partial \mathcal{R}_{J,a}}{\partial J_b^{n+1}} &= -\frac{1}{2}\,\Delta t \left(\tau_J\; V_b\; \frac{1}{c_p} \frac{\kappa}{\rho}\; N_{ab}\; \tilde{\nabla}_0 \tilde{W}_a(X_b) \right)
\end{aligned} \tag{5.20}$$

Lastly, the expression for $\mathcal{R}_{x,a}$, for which the derivatives are trivial, is

$$\mathcal{R}_{x,a} = x_a^{n+1,k} - x_a^n - \frac{1}{2}\,\Delta t \left(v_a^{n+1} + v_a^n \right) \tag{5.21}$$

If the linear reconstruction procedure is employed, the derivatives in (5.18) and (5.20) must include additional terms. Refer back to (3.53), then the interface velocity v_f can be written as

$$v_f^- = v_a + \left(\sum_{b\in\Lambda_a} 1 \right)^{-1} \frac{v_b - v_a}{\parallel d_{ab} \parallel} N_{ab} \tag{5.22}$$

then the corresponding derivatives with respect to v becomes

$$\frac{\partial v_f^-}{\partial v_a} = 1 - \left(\sum_{b\in\Lambda_a} 1 \right)^{-1} \frac{1}{\parallel d_{ab} \parallel} N_{ab} \quad \text{and} \quad \frac{\partial v_f^-}{\partial v_b} = \left(\sum_{b\in\Lambda_a} 1 \right)^{-1} \frac{1}{\parallel d_{ab} \parallel} N_{ab} \tag{5.23}$$

The same observation applies to $\sigma_f^{+,-}$ and its derivatives with respect to J.
The derivatives computed above allow to build row by row the Jacobian matrix needed for Newton-Raphson iteration in (5.16). In this specific tension problem, the Jacobian

matrix, as seen in (5.18) and (5.20), is independent of the problem variables, namely v and J. As a result, it is computed once at each time step, outside the loop, differently from what is shown in the algorithm 2.
Another important feature of this problem is its linearity, which is preserved by the linear elastic constitutive law. For this class of problems, the Newton-Raphson scheme is expected to converge in a single iteration, as shown also in [6, 28], where the same tension problem is solved using the finite element method (FEM).
Verifying convergence in a single iteration serves as a validation of the Jacobian computation, since the Newton-Raphson method behaves as expected if the Jacobian corresponds exactly to the derivative of the residual function.

5.2 1D test problem

To verify the proposed implicit integration scheme, the algorithm 2 is applied to a 1D beam under tension (figure 5.1). The governing equations are given by (5.15) and the material is assumed to be linear elastic as in (5.5).

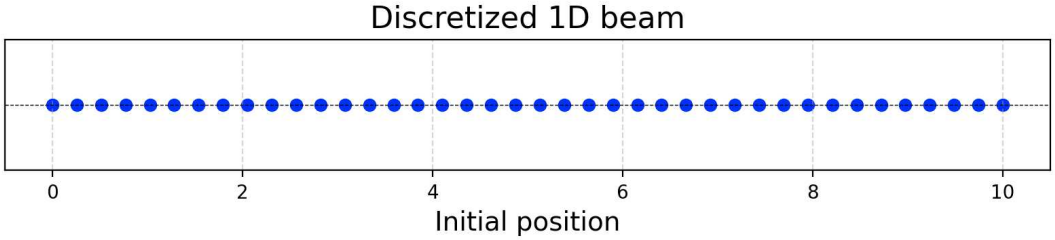


Figure 5.1: SPH discretization of a 1D beam. The beam, in this example, has a total length of $L = 10$ and is discretized using 40 equally spaced particles.

To fully define the problem, boundary conditions must be imposed and initial condition defined. In this case, the left extrema is clamped and the corresponding particles cannot move in horizontal direction[4].
Initially, consider the system in absence of external forces and with the following initial velocity

$$v_a^{start} = \sin \frac{\pi}{2L} x$$

this initial velocity is smooth and does not present shocks. In figure 5.2, the problem is solved with explicit and implicit SPH and compared with FEM solution. The time step used in the implicit scheme is $\Delta t = 0.01$ while $\Delta t = 0.001$ is adopted in the explicit scheme. The three curves plotted, overlaps.
Note that given this initial sinusoidal velocity distribution the stabilization terms are not required. Indeed, these terms are typically dissipative and are introduced to suppress spurious high-frequency energy modes. Although stabilization could still be applied in this context, it would introduce energy dissipation into the system leading to an inaccurate solution.
At this point, it is crucial to recall the limitation of the explicit scheme. As stated by the CFL condition (4.3), the maximum time step allowed depends not only on the material properties such as c_p, and the time integration parameter α_{CFL} but also on the spatial discretization.

[4]The boundary conditions are enforced by setting the corresponding row and column of the Jacobian matrix to zero, except for the diagonal element, which is set to one for the clamped particle. The corresponding value in the residual vector is also set to zero.

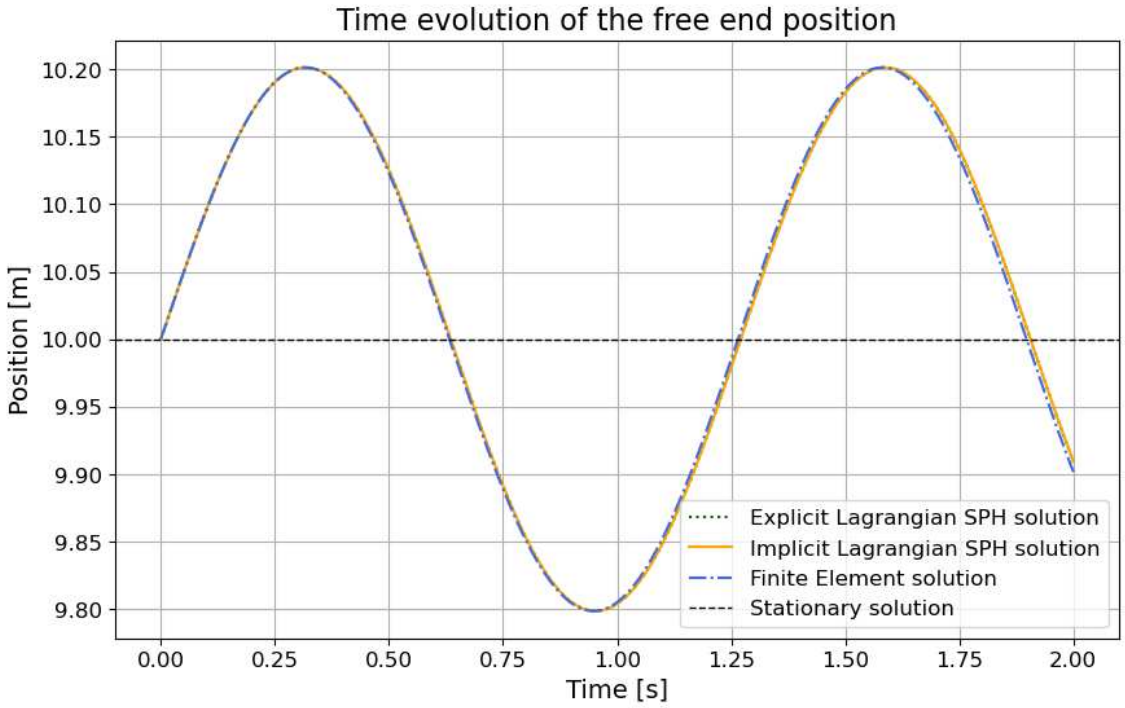


Figure 5.2: **1D tension**: Comparison of the results obtained with non-stabilized Lagrangian SPH (40 particles) and standard FEM (40 elements). The elastic constant and the beam length are set respectively to $\kappa = 1e^6$ and $L = 10\ m$. The plot shows also the stationary solution.

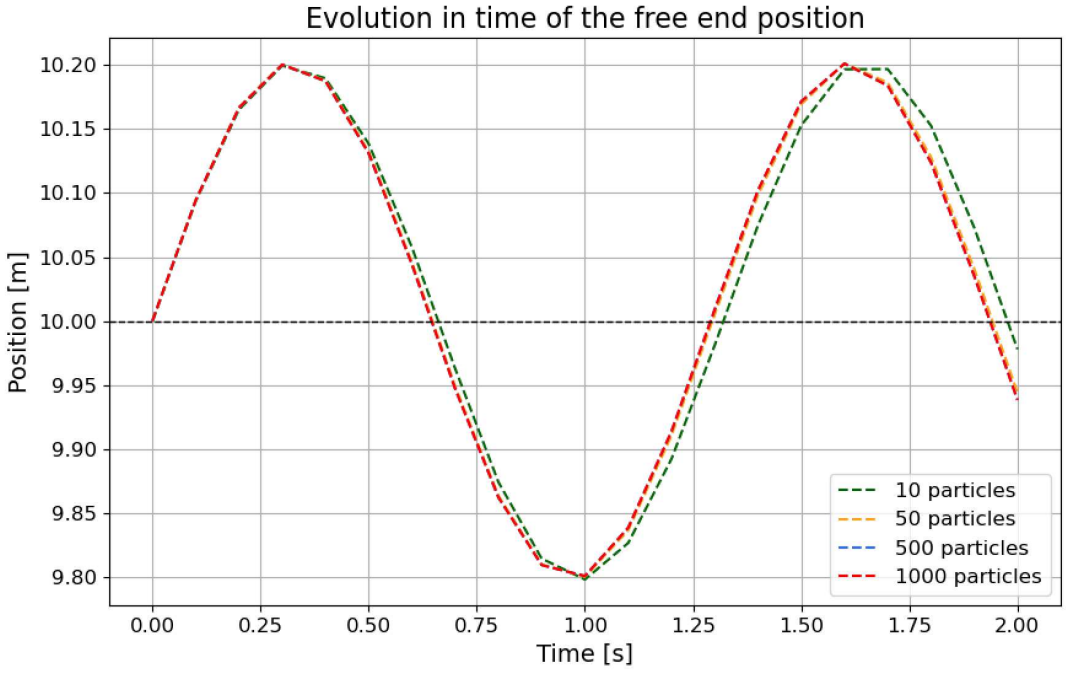


Figure 5.3: **1D tension**: Comparison of the results obtained with non-stabilized implicit Langrangian SPH using four different refining meshes. The elastic constant and the beam length are set respectively to $\kappa = 1e^8$ and $L = 10\ m$.

In general, a finer mesh provides a better approximate solution, assuming that the numer-

ical solver behaves as expected. However, when using explicit integration in SPH, a finer discretization, for this example, leads to prohibitively small time steps, as shown in Table 5.1

n particles	d [m]	Δt [s]
10	1	$\approx 1 \times 10^{-2}$
50	0.2	$\approx 2 \times 10^{-3}$
500	0.02	$\approx 2 \times 10^{-4}$
1000	0.01	$\approx 1 \times 10^{-4}$

Table 5.1: Time steps Δt required by the explicit scheme for refining discretizations. The fixed values of α_{CFL} and κ are respectively 0.3 and 10^6.

On the other hand, when using an implicit time integration scheme, the time step no longer depends by the mesh size or by the material properties. In Figure 5.3, the problem is solved implicitly using different mesh resolutions while maintaining a fixed time step of $\Delta t = 0.1$. The results show that the solution curves converge to a single profile as the number of particles increases. This confirms, that in the proposed implicit scheme the time step is independent of the spatial discretization. The only limitation on Δt is related to the dynamic behavior of the solution itself. If the time step is not sufficiently smaller than the characteristic oscillation period, the method, in this case, may not accurately capture the oscillations of the system.

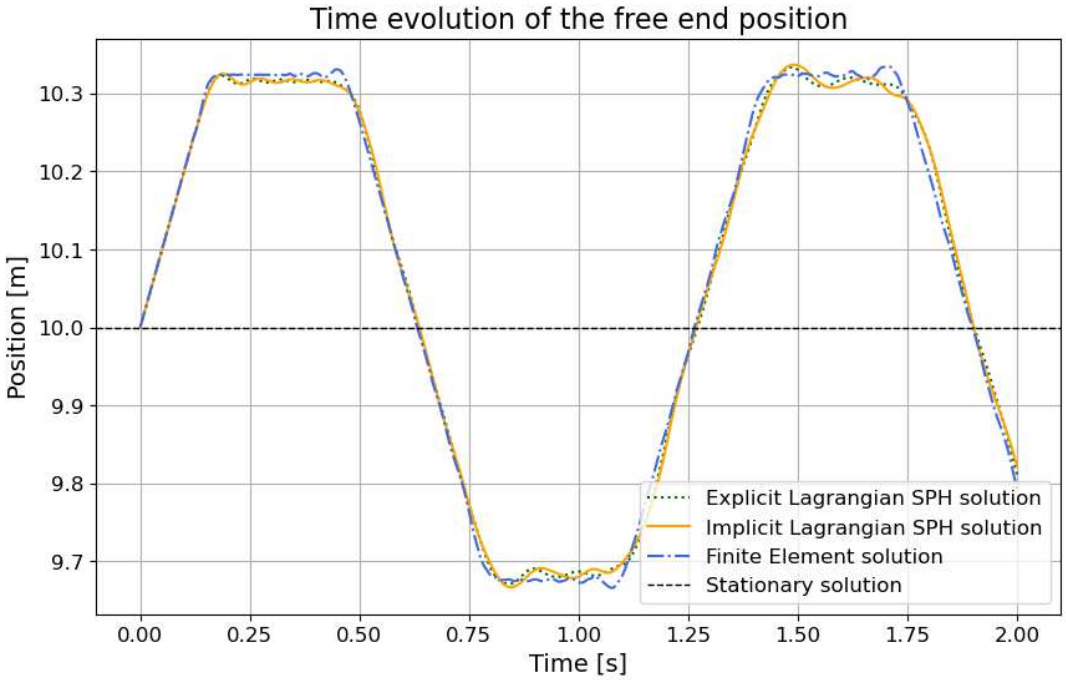


Figure 5.4: **1D tension**: Comparison of the results obtained with non-stabilized Lagrangian SPH (40 particles) and standard FEM (40 elements). The elastic constant and the beam length are set respectively to $\kappa = 1e^6$ and $L = 10\ m$. The applied initial velocity presents a shock.

In order to evaluate the influence of stabilization, a shock must be introduced into the system. Also this time no external forces are considered but, the following initial velocity

field is adopted

$$v_a^{start} = 0 \quad \forall a \in \left\{0, \dots, \frac{nn}{2} - 1\right\} \quad \text{and} \quad v_{start} = 2 \quad \forall a \in \left\{\frac{nn}{2}, \dots, nn\right\}$$

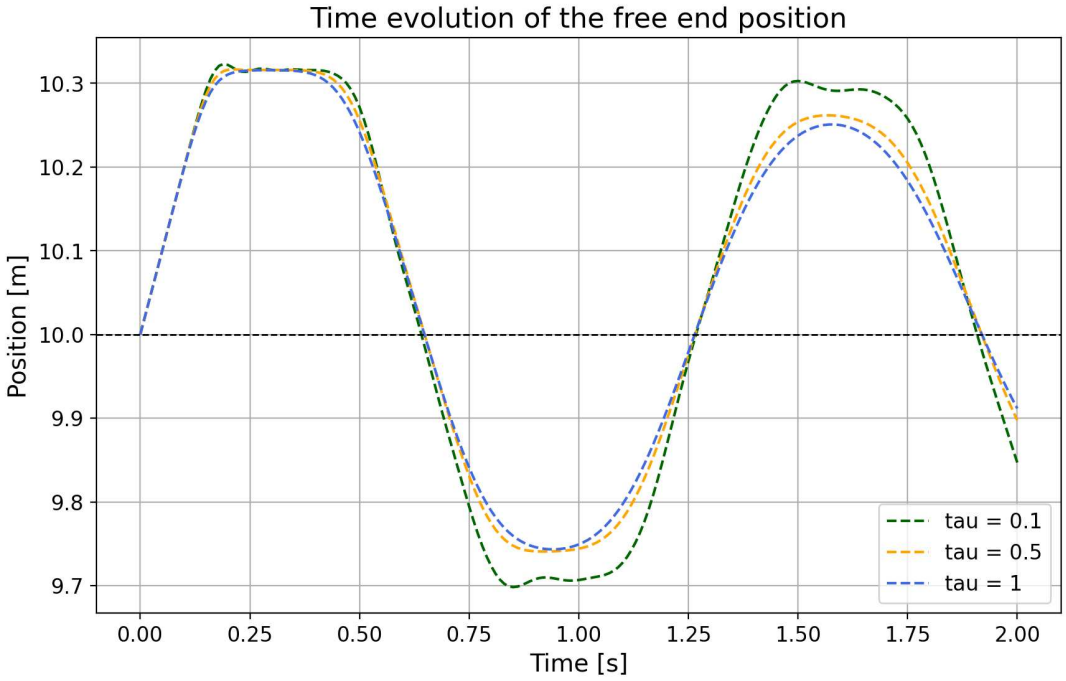


Figure 5.5: **1D tension**: Results obtained with stabilized Langrangian SPH and implicit time integration scheme. Different values of τ_v and τ_J are compared.

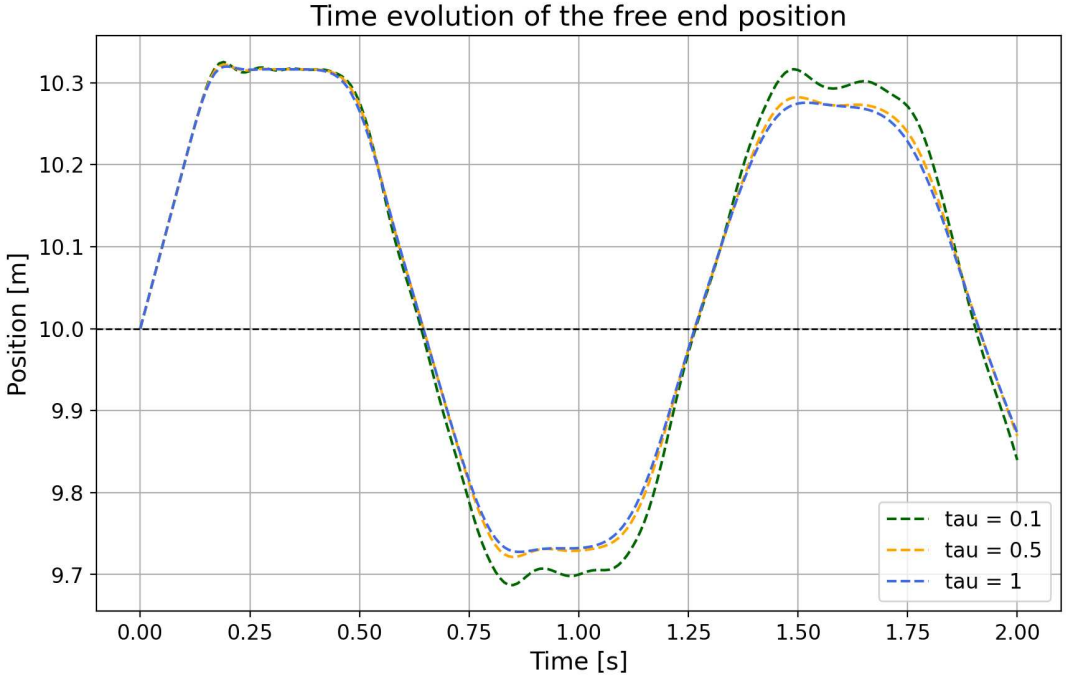
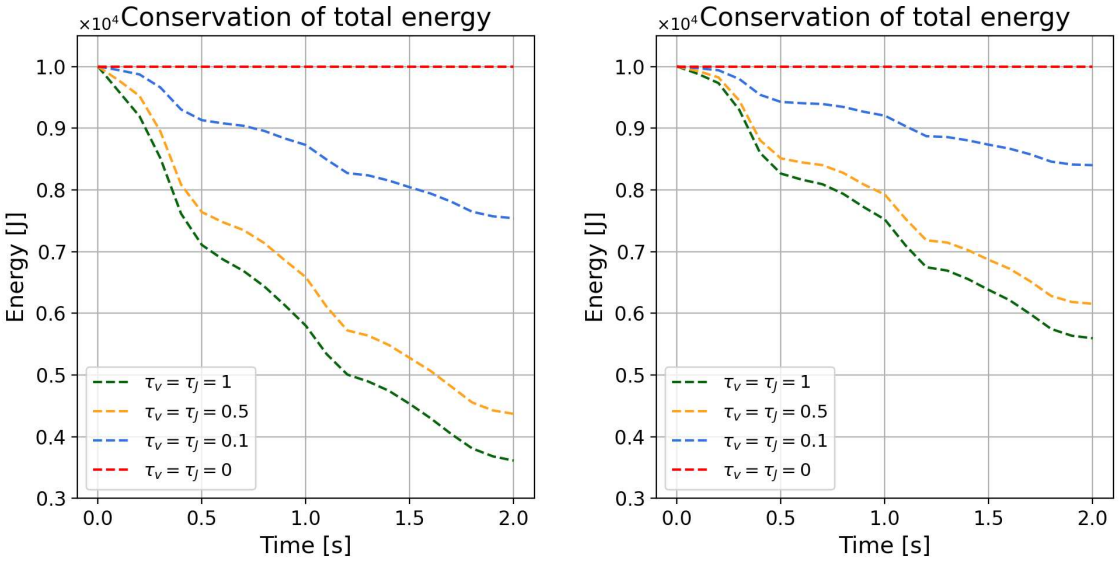


Figure 5.6: **1D tension**: Results obtained with stabilized Langrangian SPH and implicit time integration scheme. The linear reconstruction procedure described in Chapter 3 is adopted. Different values of τ_v and τ_J are compared.

In this example half of the beam is initially moving with velocity $v \neq 0$, this configuration introduce a shock in the solution that needs to be correctly treated as visible in Figure 5.4.
In particular, the solution exhibits small spurious oscillations superimposed on the wave peaks. The time step chosen for the implicit scheme is $\Delta t = 0.01$[5].
Figures 5.5 and 5.6 present the simulation results for three different values of $\tau_v = \tau_J = \{1,\ 0.5,\ 0.1\}$. A higher value of τ, corresponding to stronger stabilization, leads to a faster smoothing of the solution.
In Figure 5.6, the linear reconstruction procedure is employed, enabling a linear approximation of the discontinuity between two neighboring particles. This approach reduces the influence of the stabilization term while still ensuring effective smoothing. In both cases, a smooth solution is achieved for all three values of τ.
Since stabilization introduces numerical dissipation, its use should be restricted to regions where it is actually needed[6], and the parameter τ must be chosen appropriately.
Given the known initial conditions, the initial kinetic energy of the system can be computed analytically as

$$K = \frac{1}{2}mv^2 = 0.5 \cdot 5 \cdot 1000 \cdot 2^2 = 10\ 000\ J \tag{5.24}$$

assuming unitary cross-sectional area, homogeneous material with density $\rho_0 = 1000$ and the velocity profile is the one presented above. In this specific case, the initial kinetic energy corresponds to the total energy, while in general the latter is the sum of kinetic, internal and potential energy. During the simulation, part of this kinetic will periodically switch into potential energy.



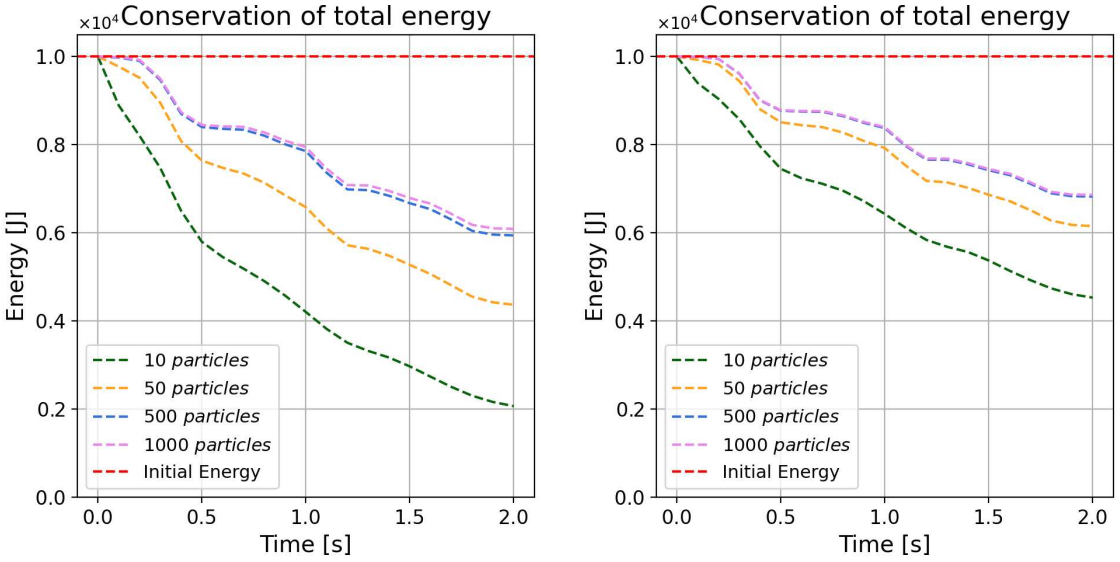
(a) Without linear reconstruction procedure (b) With linear reconstruction procedure

Figure 5.7: Evolution of system's energy in time comparing the two stabilization strategies for different values of τ.

Figure 5.7 shows the energy of the system during the simulation in time using the stabilized schemes. As expected, increasing τ leads to higher energy dissipation, while looking

[5]With a bigger time step, the scheme would not be able to capture the unstable modes.
[6]This behavior is encouraged by the fact that the stabilization term is proportional to the jump between neighboring particles, which tends to localize its effect near discontinuities.

at the right graph, the linear reconstruction procedure helps reduce it. In the absence of stabilization, the total energy remains constant throughout the simulation and its value coincides with the one in (5.24).
In Figure 5.8 the values of τ are kept constant while the number of particles increases. The graphs show that as the discretization refines the loss of energy decreases. Also in this case the reconstruction procedure allows a minor loss of energy.



(a) Without linear reconstruction procedure

(b) With linear reconstruction procedure

Figure 5.8: Evolution of system's energy in time comparing the two stabilization strategies while refining the mesh. The values of τ_v and τ_J are constant and equal to 0.5.

Chapter 6

Numerical examples: Extension to multi-dimensional problems

This chapter extend the analysis to multi-dimensional case.
After recalling the full Lagrangian SPH formulation (3.39), the focus is on computing the full derivatives of the residuals required by the Newton-Raphson method. Finally, a series of 2D numerical examples are presented in order to assess the robustness, effectiveness and applicability of the new SPH computational framework. The results obtained are compared with an adapted version of the explicit SPH code written by Lee et al. [17], which was tested in [22].

6.1 Multi-dimensional problem and Jacobian construction

The derivation of the problem in multiple dimensions has already been presented in Chapter 3. The following equations present the final SPH discretized formulation

$$\begin{aligned} \rho\frac{\partial \boldsymbol{v}_a}{\partial t} &= -\sum_{b\in\Lambda_a} V_b\ \boldsymbol{P}_b\tilde{\nabla}_0\tilde{W}_a(\boldsymbol{X}_b) + \boldsymbol{b}_a + \sum_{b\in\Lambda_a}\boldsymbol{\mathcal{D}}_{\boldsymbol{v}}^{ab} \\ \frac{\partial J_a}{\partial t} &= \boldsymbol{H}_a : \sum_{b\in\Lambda_a} V_b\ \boldsymbol{v}_b\otimes\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a) + \sum_{b\in\Lambda_a}\mathcal{D}_p^{ab} \end{aligned} \tag{6.1}$$

where the stabilization terms are defined as

$$\begin{aligned} \boldsymbol{\mathcal{D}}_{\boldsymbol{v}}^{ab} &= \tau_{\boldsymbol{v}}\sum_{b\in\Lambda_a} V_b\left\{\left[c_p(\boldsymbol{n}_{ab}\otimes\boldsymbol{n}_{ab})\ (\boldsymbol{v}_f^+ - \boldsymbol{v}_f^-)\right]\otimes\boldsymbol{N}_{ab}\right\}\cdot\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)^{Ave} \\ \mathcal{D}_p^{ab} &= \tau_p\sum_{b\in\Lambda_a} V_b\ \left[\frac{1}{\rho c_p}(\boldsymbol{n}_{ab}\otimes\boldsymbol{n}_{ab})\left((\boldsymbol{P}_f^+ - \boldsymbol{P}_f^-)\boldsymbol{N}_{ab}\right)\right]\cdot\left[H_a\tilde{\nabla}_0\tilde{W}_b(\boldsymbol{X}_a)\right]^{Ave} \end{aligned} \tag{6.2}$$

Recall the elastic constitutive law used in Chapter 3 to close the system of equations

$$\boldsymbol{P} = -p\boldsymbol{H} = -p\ J\boldsymbol{F}^{-T} \tag{6.3}$$

where only volumetric contribution is considered. Note that, differently from 1D, the Piola stress cannot be simplified and depends on both $\boldsymbol{v}$ and J.
The time integration of the multidimensional problem using an implicit scheme follows the same approach used for the 1D case. The Newton-Raphson method is applied to solve

$\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}) = 0$, where residuals are defined using (6.1) as

$$\begin{aligned}
\boldsymbol{\mathcal{R}}_{\boldsymbol{v}}(\boldsymbol{\mathcal{U}}^{n+1},\boldsymbol{\mathcal{U}}^{n}) &= \rho\ \boldsymbol{v}_a^{n+1,k} - \rho\ \boldsymbol{v}_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{\boldsymbol{v}}_a^{n+1,k} + \dot{\boldsymbol{v}}_a^{n}\right) = 0 \\
\boldsymbol{\mathcal{R}}_{J}(\boldsymbol{\mathcal{U}}^{n+1},\boldsymbol{\mathcal{U}}^{n}) &= J_a^{n+1,k} - J_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{J}_a^{n+1,k} + \dot{J}_a^{n}\right) = 0 \\
\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(\boldsymbol{\mathcal{U}}^{n+1},\boldsymbol{\mathcal{U}}^{n}) &= \boldsymbol{x}_a^{n+1,k} - \boldsymbol{x}_a^{n} - \Delta t\ \frac{1}{2}\ \left(\dot{\boldsymbol{x}}_a^{n+1,k} + \dot{\boldsymbol{x}}_a^{n}\right) = 0
\end{aligned} \tag{6.4}$$

while the linear system to solve is

$$\begin{bmatrix}
\frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial \boldsymbol{x}} \\
\frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{J}}{\partial \boldsymbol{x}} \\
\frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial J} & \frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}}{\partial \boldsymbol{x}}
\end{bmatrix}
\begin{bmatrix}
\Delta \boldsymbol{v} \\ \Delta J \\ \Delta \boldsymbol{x}
\end{bmatrix} = -
\begin{bmatrix}
\boldsymbol{\mathcal{R}}_{\boldsymbol{v}} \\ \boldsymbol{\mathcal{R}}_{J} \\ \boldsymbol{\mathcal{R}}_{\boldsymbol{x}}
\end{bmatrix} \tag{6.5}$$

The derivatives of these residual terms become significantly more complex in multi-D. Moreover, the problem is no longer linear, meaning that the Jacobian matrix needs to be computed at each Newton-Raphson iteration.
The computation of the derivatives requires repeated application of the chain rule, making the process lengthy and complex. Only the derivatives of the most critical and significant steps will be presented in the following pages.
First of all, recall that the deformation gradient can be written in terms of the velocity field. Indeed, the positions $\boldsymbol{x}$ are updated over time using the Crank–Nicolson scheme

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + \frac{1}{2}\Delta t(\boldsymbol{v}^{n+1} + \boldsymbol{v}^{n}) \tag{6.6}$$

then the update formula for the deformation gradient tensor will be

$$\boldsymbol{F}^{n+1} = \frac{\partial \boldsymbol{x}^{n+1}}{\partial \boldsymbol{X}} = \boldsymbol{F}^{n} + \frac{1}{2}\Delta t\left(\dot{\boldsymbol{F}}^{n+1} + \dot{\boldsymbol{F}}^{n}\right) \tag{6.7}$$

At this point it is trivial to recognize that $\boldsymbol{F}$ depends on $\boldsymbol{v}$, then the derivative of $\boldsymbol{F}^{-T}$ with respect to $\boldsymbol{v}$ (present in the Piola stress terms) must be computed. To do that, first of all, state that

$$\frac{\partial \boldsymbol{F}^{-T}}{\partial \boldsymbol{v}} = -\boldsymbol{F}^{-T}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\right)^{T}\boldsymbol{F}^{-T} \tag{6.8}$$

Then, the derivative of $\boldsymbol{F}_a^{n+1}$ with respect to $\boldsymbol{v}_b^{n+1}$ is needed.

$$\frac{\partial \boldsymbol{F}_a^{n+1}(\boldsymbol{v}_b^{n+1}, \boldsymbol{v}_b^{n})}{\partial \boldsymbol{v}_b^{n+1}} = \frac{\partial}{\partial \boldsymbol{v}_b^{n+1}}\left(\boldsymbol{F}_a^{n} + \frac{1}{2}\Delta t\left(\dot{\boldsymbol{F}}_a^{n+1}(\boldsymbol{v}_b^{n+1}) + \dot{\boldsymbol{F}}_a^{n}(\boldsymbol{v}_b^{n})\right)\right) \tag{6.9}$$

Note that only $\dot{\boldsymbol{F}}^{n+1}$ depends on $\boldsymbol{v}^{n+1}$, thus

$$\begin{aligned}
\frac{\partial \boldsymbol{F}_a^{n+1}(\boldsymbol{v}_b^{n+1}, \boldsymbol{v}_b^{n})}{\partial \boldsymbol{v}_b^{n+1}} &= \frac{1}{2}\Delta t\frac{\partial \dot{\boldsymbol{F}}_a^{n+1}(\boldsymbol{v}_b^{n+1})}{\partial \boldsymbol{v}_b^{n+1}} = \frac{1}{2}\Delta t\frac{\partial}{\partial \boldsymbol{v}_b^{n+1}}\left(\sum_{b\in\Lambda_a} V_b \boldsymbol{v}_b^{n+1} \otimes \nabla_0 W_b(\boldsymbol{X}_a)\right) \\
&= \frac{1}{2}\Delta t\frac{\partial}{\partial \boldsymbol{v}_b^{n+1}}\sum_{b\in\Lambda_a} V_b \begin{bmatrix} v_{b,x}\ \nabla_0 W_b(\boldsymbol{X}_a)_x & v_{b,x}\ \nabla_0 W_b(\boldsymbol{X}_a)_y \\ v_{b,y}\ \nabla_0 W_b(\boldsymbol{X}_a)_x & v_{b,y}\ \nabla_0 W_b(\boldsymbol{X}_a)_y \end{bmatrix}
\end{aligned} \tag{6.10}$$

The final result is the following third order tensor

$$\frac{\partial \boldsymbol{F}_a^{n+1}}{\partial \boldsymbol{v}_b^{n+1}} = \frac{1}{2}\Delta t\ V_b\left[\begin{bmatrix} \nabla_0 W_b(\boldsymbol{X}_a)_x & \nabla_0 W_b(\boldsymbol{X}_a)_y \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ \nabla_0 W_b(\boldsymbol{X}_a)_x & \nabla_0 W_b(\boldsymbol{X}_a)_y \end{bmatrix}\right] \tag{6.11}$$

Remark 7. To see how to compute the derivative of $\boldsymbol{F}^{-T}$ with respect to the velocity field, consider

$$\frac{\partial}{\partial \boldsymbol{v}}\left(\boldsymbol{F}\boldsymbol{F}^{-1}\right) = 0 \quad \rightarrow \quad \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\boldsymbol{F}^{-1} + \boldsymbol{F}\frac{\partial \boldsymbol{F}^{-1}}{\partial \boldsymbol{v}} = 0 \tag{6.12}$$

then pre-multiply equation (6.12)b by $\boldsymbol{F}^{-1}$ the following relation is obtained

$$\frac{\partial \boldsymbol{F}^{-1}}{\partial \boldsymbol{v}} - \boldsymbol{F}^{-1}\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\boldsymbol{F}^{-1} = 0 \tag{6.13}$$

Recall that the transposition does not affect the dependence from the velocity. Using matrix property $(ABC)^T = C^T B^T A^T$ the expression for the derivative is

$$\frac{\partial \boldsymbol{F}^{-T}}{\partial \boldsymbol{v}} = -\boldsymbol{F}^{-T}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\right)^T \boldsymbol{F}^{-T} \tag{6.14}$$

The attention is now turned to stabilization terms. These terms include normal vectors $\boldsymbol{n}_{ab}$ and shock wave speed c_p.
Consider the normal vector $\boldsymbol{n}_{ab}$ between two particle a and b defined as

$$\boldsymbol{n}_{ab} = \frac{\boldsymbol{x}_b - \boldsymbol{x}_a}{\parallel \boldsymbol{x}_b - \boldsymbol{x}_a \parallel} \tag{6.15}$$

Letting $\boldsymbol{r} = \boldsymbol{x}_b - \boldsymbol{x}_a$ the derivative with respect to $\boldsymbol{x}_b$ is

$$\begin{aligned}\frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{x}_b} &= \frac{\partial}{\partial \boldsymbol{x}_b}\left(\frac{\boldsymbol{r}}{\parallel \boldsymbol{r} \parallel}\right)\frac{\partial \boldsymbol{r}}{\boldsymbol{x}_b} = \left(\frac{\mathbb{I}}{\parallel \boldsymbol{r} \parallel} - \frac{\boldsymbol{r}\boldsymbol{r}^T}{\parallel \boldsymbol{r} \parallel^3}\right) \\ &= \frac{1}{\parallel \boldsymbol{r} \parallel}\left(\mathbb{I} - \frac{\boldsymbol{r}\boldsymbol{r}^T}{\parallel \boldsymbol{r} \parallel^2}\right)\end{aligned} \tag{6.16}$$

hence

$$\frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{x}_b} = \frac{1}{\parallel \boldsymbol{r} \parallel}\ (\mathbb{I} - \boldsymbol{n} \ \otimes \ \boldsymbol{n}) \tag{6.17}$$

Finally, the derivative of the outer product $\boldsymbol{n} \otimes \boldsymbol{n}$ with respect to the position $\boldsymbol{x}_b$ is computed using chain rule

$$\frac{\partial}{\partial \boldsymbol{x}_b}(\boldsymbol{n} \otimes \boldsymbol{n}) = \left(\frac{\partial \boldsymbol{n}}{\partial \boldsymbol{x}_b}\right) \otimes \boldsymbol{n} + \boldsymbol{n} \otimes \left(\frac{\partial \boldsymbol{n}}{\partial \boldsymbol{x}_b}\right) \tag{6.18}$$

Consider now the speed of shock propagation c_p, present in both stabilization terms. In general, this variable cannot be considered constant, as it changes when the volume associated with a particle deforms, following the relation

$$c_p = \frac{c_{p,0}}{\sqrt{\lambda_{min}}} \tag{6.19}$$

where λ_{min} is the minimum eigenvalue of the Cauchy-Green strain tensor. Since this tensor depends on $\boldsymbol{F}$ and consequently on $\boldsymbol{v}$, c_p must be derived during Jacobian construction. First of all, recall the definition of eigenvalue

$$\lambda_{min} = \boldsymbol{u}_{min}^T\ \boldsymbol{C}\ \boldsymbol{u}_{min} \tag{6.20}$$

The only term depending on $\boldsymbol{v}$ is the tensor $\boldsymbol{C}$ that is derived as

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}} = \frac{\partial}{\partial \boldsymbol{v}}\left(\boldsymbol{F}^T\boldsymbol{F}\right) = \left(\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\right)^T \boldsymbol{F} \ + \ \boldsymbol{F}^T\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\right)\right) \tag{6.21}$$

Note that the derivative of the eigenvector $\boldsymbol{u}_{min}$ is neglected. Finally, the derivative of $\sqrt{\lambda_{min}}$ reads as

$$\frac{\partial}{\partial \boldsymbol{v}} \frac{c_p}{\sqrt{\lambda_{min}}} = -\frac{1}{2} c_p (\lambda_{min})^{\frac{3}{2}} \cdot \boldsymbol{u}_{min}^T \left(\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}} \right)^T \boldsymbol{F} \; + \; \boldsymbol{F}^T \left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}} \right) \right) \boldsymbol{u}_{min} \tag{6.22}$$

At this point is possible to compute the derivatives of the residuals functions in (6.4) and determine block by block the Jacobian matrix of the Newton-Raphson scheme in (6.5).

Remark 8. As mentioned before, the derivatives in the multi-D problem are lengthy and not trivial, requiring repeated application of the chain rule. Therefore, verifying the correctness of the Jacobian becomes of primary importance once the full residual derivative is introduced. A straightforward approach is to verify the Jacobian's correctness by checking if the numerical scheme exhibits second-order convergence during the simulation. Another practical approach is based on Taylor's expansion; this technique results accurate and faster since it can be implemented separately.
Recall that the Jacobian matrix of a residual vector $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}) \in \mathbb{R}^n$ with respect to the vector of unknowns $\mathcal{U} \in \mathbb{R}^n$ is defined as:

$$\mathcal{J}_{ij} = \frac{\partial \mathcal{R}_i}{\partial \mathcal{U}_j} \tag{6.23}$$

where each column of $\boldsymbol{\mathcal{J}}$ tells how the whole residual changes if a single component of the vector of unknowns is perturbed. By computing the Taylor expansion of $\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1})$ around an arbitrary perturbation vector[1] $\boldsymbol{\delta}$, the following relation is obtained

$$\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1} + \epsilon \boldsymbol{\delta}) = \boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1}) + \epsilon \; \boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{U}}^{n+1}) \cdot \boldsymbol{\delta} + \mathcal{O}(\epsilon^2) \tag{6.24}$$

where ϵ is an arbitrary coefficient that controls the entity of the perturbation. Then, truncation error is defined as

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{R}}_i(\boldsymbol{\mathcal{U}}^{n+1} + \epsilon \boldsymbol{\delta}) - \boldsymbol{\mathcal{R}}_i(\boldsymbol{\mathcal{U}}^{n+1}) - \epsilon \; \boldsymbol{\mathcal{J}}(\boldsymbol{\mathcal{U}}^{n+1}) \cdot \boldsymbol{\delta} = \mathcal{O}(\epsilon^2) \tag{6.25}$$

Since the error is of order $\mathcal{O}(\epsilon^2)$, it should decrease as the magnitude of the perturbation ϵ becomes smaller. Moreover the truncation $\boldsymbol{\varepsilon}$ error should satisfy the following expression for all different values of ϵ

$$\| \; \boldsymbol{\varepsilon} \; \| = C \; \epsilon^2 \tag{6.26}$$

where C is a constant.
Table 6.1 presents a test performed in a square domain, where quadratic convergence is verified. This confirms the correctness of the computed Jacobian.

ϵ	$\| \; \varepsilon \; \|$	$\| \; \varepsilon \; \| \; / \epsilon^2$
1×10^{-2}	4.473×10^{1}	4.473×10^{3}
1×10^{-3}	4.473×10^{-1}	4.473×10^{3}
1×10^{-4}	4.473×10^{-3}	4.473×10^{3}
1×10^{-5}	4.473×10^{-5}	4.473×10^{3}
1×10^{-6}	4.473×10^{-7}	4.473×10^{3}

Table 6.1: Test performed within a square domain discretized with 30 particles

---

[1] The perturbation vector is randomly generated and then is normalized to avoid numerical instabilities

6.2 Numerical tests: Elastic material

As anticipated, in this section a series of numerical examples are presented to verify the applicability and correctness of the code proposed, by comparing the existing explicit scheme with the novel implicit one. All simulations are restricted to inviscid flow dynamics without severe topological changes in the free surface.

Piston test

In this example, a rectangular domain $\Omega = 3 \times 2.5\ m$ is discretized using 100 particles.

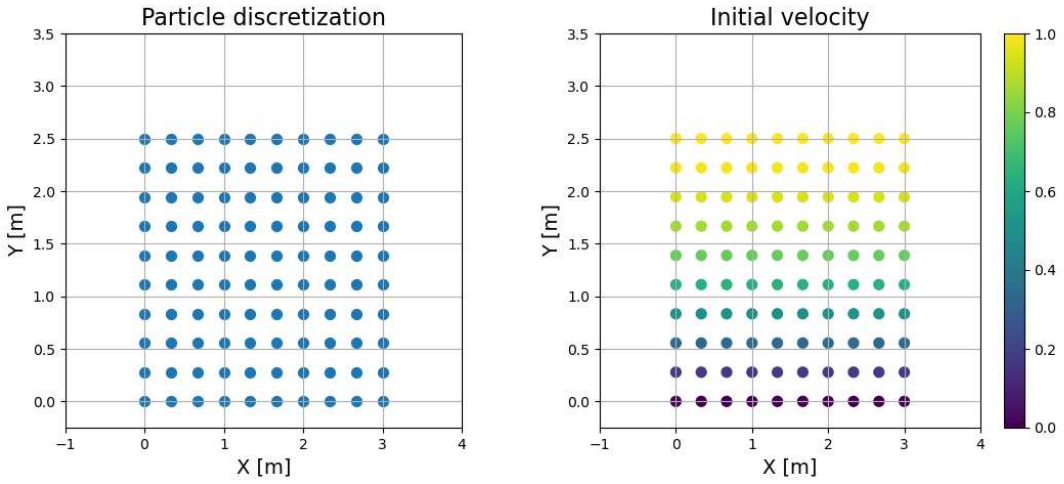


Figure 6.1: Domain discretization and initial velocity field

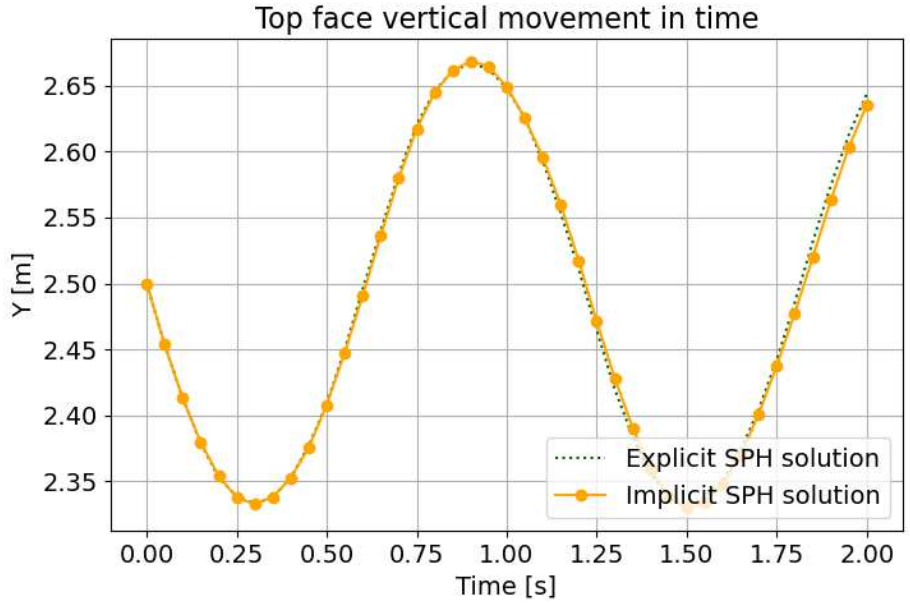


Figure 6.2: Vertical movement of the upper face during the simulation. Comparison between implicit and explicit time integration scheme.

All particles are fixed in the horizontal direction, so that only vertical displacements are allowed. External forces, including gravity, are not considered while a sinusoidal initial

velocity is applied in the negative y-direction, defined as:

$$v_a^{start} = \sin \frac{\pi}{2H} y_a \tag{6.27}$$

where $H = y_{max} - y_{min}$.
The material properties under consideration are density equal to $\rho_0 = 1000\ kg/m^3$, Young modulus $E = 1 \times 10^6\ Pa$ and Poisson coefficient $\nu = 0.3$.[2]

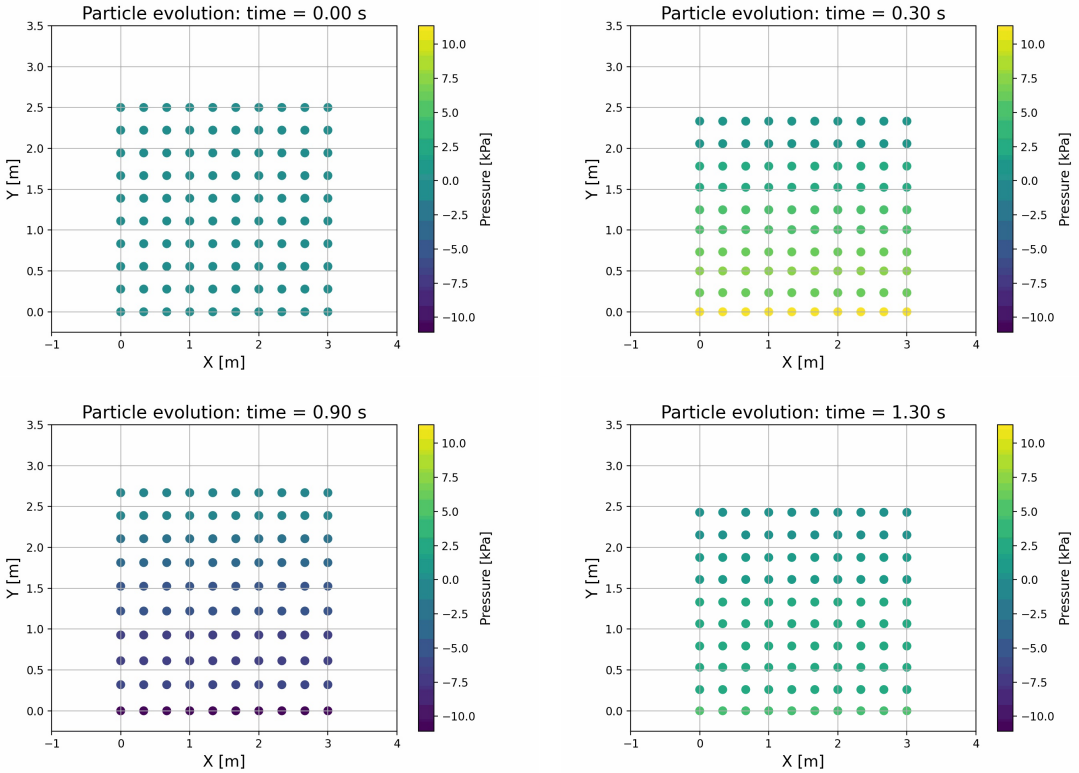


Figure 6.3: Particle position at different time step during the simulation. The plots include a color bar visualizing the pressure.

Figure 6.2 shows the vertical displacement of the upper surface over time. As expected, the material exhibits an oscillatory and symmetric response. In fact, due to the absence of shear resistance and the imposed constraint on horizontal motion, the material response is purely volumetric governed by the linear elastic constitutive law. In this specific case, no stabilization techniques are required, and the results presented do not include any stabilization terms.
Figure 6.3 shows the evolution of the pressure over time. As can be seen, the highest pressure values in modulus occur when the deformation is greatest. In the absence of gravity force, there is no initial pressure at the beginning of the simulation. It is worth recalling the fluid mechanics convention adopted here: positive pressure corresponds to compression, while negative pressure indicates tension. Finally, Figure 6.4 shows the evolution of total, kinetic, and internal energy. As expected, the total energy remains constant over time, as no dissipation effects are considered. Moreover, it is observed that kinetic energy is periodically converted into internal energy[3], and vice versa as expected.

[2] In 2D, the bulk modulus is obtained as function of the Young's modulus E and the Poisson coefficient ν with the following expression: $\kappa = \frac{E}{2(1-\nu)}$

[3] The energy oscillates with a period that is one half the one of particle position.

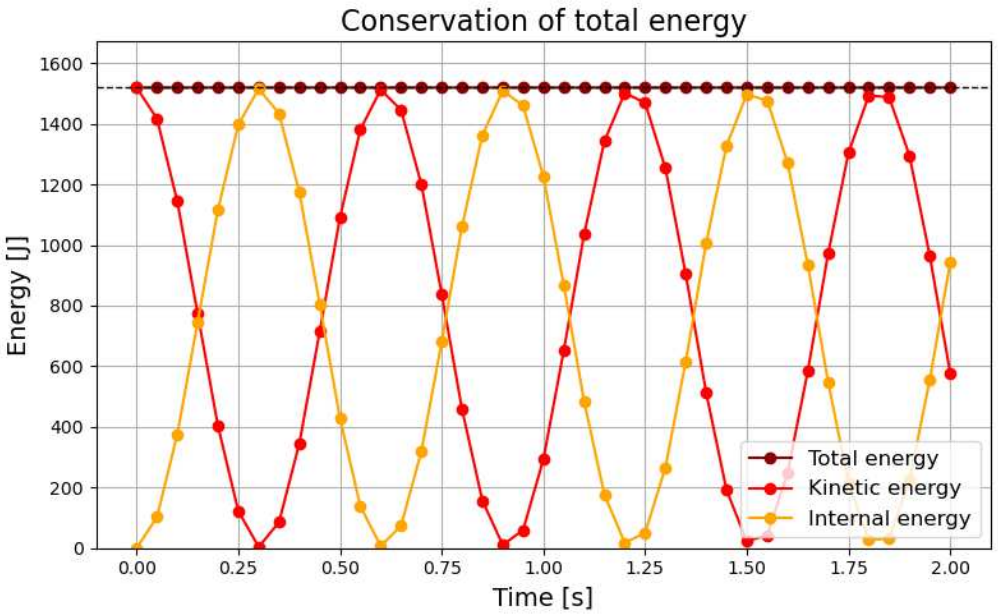


Figure 6.4: Conservation of total energy in time in absence of dissipation effects. The graph shows also the evolution of kinetic and internal energy.

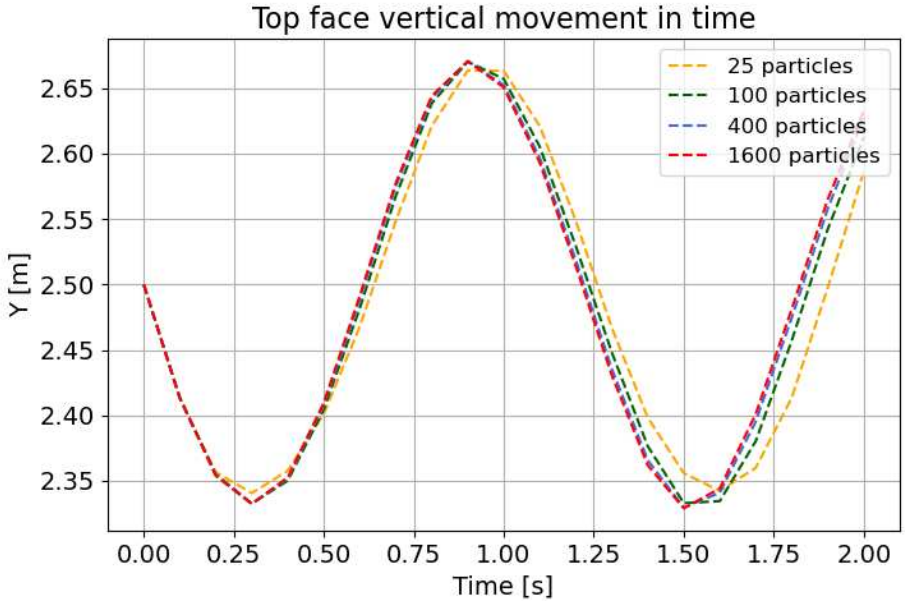


Figure 6.5: The vertical evolution in time of the upper face of the domain is compared using four different refining meshes and the novel implicit SPH scheme.

n particles	$d\ [m]$	$\Delta t\ [s]$
25	0.5	$\approx 1.8 \times 10^{-2}$
100	0.25	$\approx 8.8 \times 10^{-3}$
400	0.125	$\approx 4.4 \times 10^{-3}$
1600	0.0625	$\approx 2.2 \times 10^{-3}$

Table 6.2: Time steps Δt required by the explicit scheme for refining discretizations. The fixed values of α_{CFL}, Young's modulus E and Poisson coefficient ν are respectively 0.3, $1 \times 10^5\ Pa$ and 0.3.

As in the 1D case, the use of an explicit time integration scheme is limited by the CFL condition. Table 6.2 reports the theoretical maximum allowed Δt for different domain discretizations using an increasing number of particles.
In Figure 6.5 the time step $\Delta t = 0.1\ s$ is kept fixed while the same problem is solved using the implicit scheme for different particle resolutions. As shown, the solutions converge with increasing resolution and do not exhibit any instability issues. This shows that the time step is independent of particle refinement.

Impact test

This second example investigates the impact of an elastic fluid on a rigid surface, which coincides with the horizontal x-axis. The domain considered is $\Omega = 3 \times 5\ m$, and the discretization is obtained using a set of 225 particles. To simulate an impact, particles are not allowed to penetrate the base surface. Boundary condition are assigned by blocking the displacement in vertical direction of the particles having null y coordinate. The motion is given by a constant initial velocity $\boldsymbol{v} = [0, -1]$ and the fluid properties are density $\rho_0 = 1000\ kg/m^3$, Young's modulus $E = 1 \times 10^5\ Pa$ and Poisson coefficient $\nu = 0.3$.

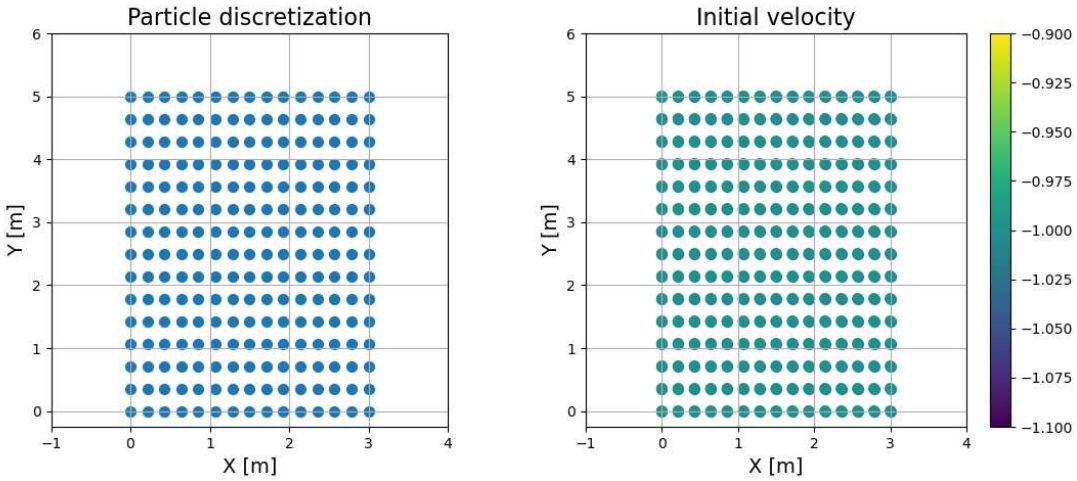


Figure 6.6: Domain discretization and initial velocity field

At the beginning of the simulation, the fluid impacts the bottom surface. This impact generates a shock wave that propagates through the material, leading to instabilities.
To improve the quality of the final solution, stabilization terms are introduced for both the velocity and pressure fields.
Figure 6.8 shows the evolution in time of the bottom right corner of the domain when using stabilization terms, highlighting the comparability of results between the two time

integration schemes. For this specific example a time step $\Delta t = 0.05\ s$ was adopted. In Figure 6.7, the evolution over time of the entire domain and the corresponding pressure distribution are illustrated. The highest pressure values are released at the beginning of the simulation, right after the impact at the center of the domain. Subsequently, the pressure decreases as the deformation spreads horizontally.
In the previous 1D tension test, where the stabilization terms defined in (5.14) were used, the velocity of shock wave propagation c_p was assumed to be constant. This assumption was reasonable, given that the material experienced only limited stretching.
However, as explained in section 3.3.6, this coefficient actually varies depending on the deformation of the material. In this example, since the material undergoes significant stretching in one direction, the value of c_p is updated, as in (3.57), and its modulus increased. Consequently, the stabilization term itself is amplified.

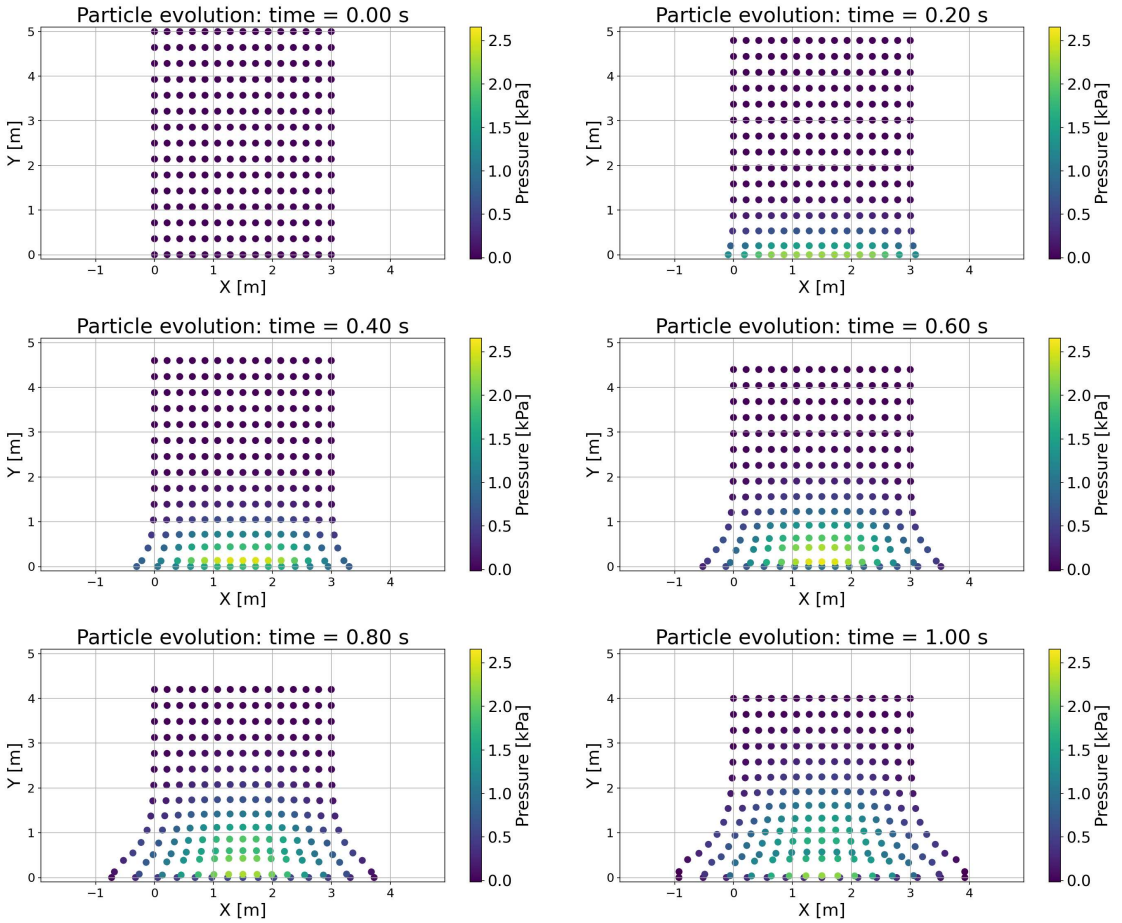


Figure 6.7: Particles position at different time step during the simulation. The plots include a color bar visualizing the pressure. The stabilization terms $\boldsymbol{\mathcal{D}_v}$ and $\mathcal{D}_p$ are $\neq 0$

However, the linear elastic constitutive law does not sufficiently penalize volume reduction. In particular, it allows for the volumetric collapse of the material at a finite energy cost.

$$\Psi(J=0) = \frac{1}{2}\kappa(J-1)^2 = \frac{1}{2}\kappa \tag{6.28}$$

This characteristic leads to instabilities in the solution, both for the explicit and implicit time integration schemes. A clear indication of this problem is the excessive deformation, concentrated at the bottom, observed at the final stage of the previous simulation in Figure 6.7.

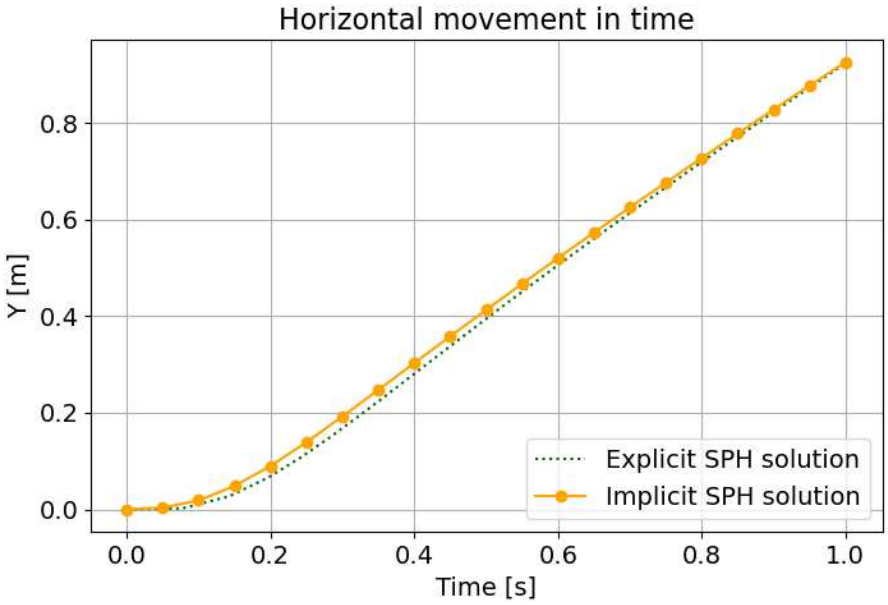


Figure 6.8: Horizontal movement of the right bottom corner during the simulation. Comparison between implicit and explicit time integration scheme.

In the implicit scheme, this problem is visible by analyzing the Jacobian matrix in Newton–Raphson iterations, where the condition number becomes extremely large, making the system unsolvable even if stabilization is applied. Due to this ill-conditioning, a breakdown occurs earlier compared to the explicit case. However, the explicit scheme is not immune; in fact, a breakdown still occurs at a later stage, driven by the same volumetric instability.
Only the results preceding the breakdown are shown in Figures 6.7 and 6.8.

6.3 Numerical test: Hyper-elastic material

To solve the issues described in the previous section, a nonlinear constitutive law for a hyper-elastic material is introduced in equation (6.1) to couple linear momentum and volume conservation equations. In particular, the following strain energy per unit of undeformed volume is defined

$$\Psi(J) = \begin{cases} \kappa(J - 1 - \frac{J^{1-\gamma}-1}{1-\gamma}), & \gamma > 1 \\ \kappa(J - 1 - \ln J), & \gamma = 1 \end{cases} \tag{6.29}$$

with $\{\gamma, \kappa\} > 0$ as positive material constant. Then, the Piola-Kirchhoff stress tensor $\boldsymbol{P}$ can be re-expressed as

$$\boldsymbol{P} = -p\boldsymbol{H}; \qquad p = -\frac{d\psi}{dJ} = \kappa\left[\left(\frac{1}{J}\right)^{\gamma} - 1\right] \tag{6.30}$$

where a positive value of the pressure indicates compression. Although this model still describes a purely volumetric response, its nonlinear behavior makes it more suitable for simulating fluids in numerical contexts. Moreover, its high resistance to compression makes it more appropriate for approximating incompressible fluids.

Remark 9. The value for the initial speed of shock propagation in this material is

$$c_{p,0} = \sqrt{\frac{\gamma\kappa}{\rho_0}} \tag{6.31}$$

obtained by evaluating the nonlinear pressure wave speed c_p at the initial configuration (see appendix A in [22]).

In the next paragraphs, the same tests of the previous section are performed with the new hyper-elastic material.

Piston test

Recall the first example and consider the domain $\Omega = 3 \times 2.5\ m$, discretized in 100 particles (as in Figure 6.1). A time step of $\Delta t = 0.1\ s$ was adopted.

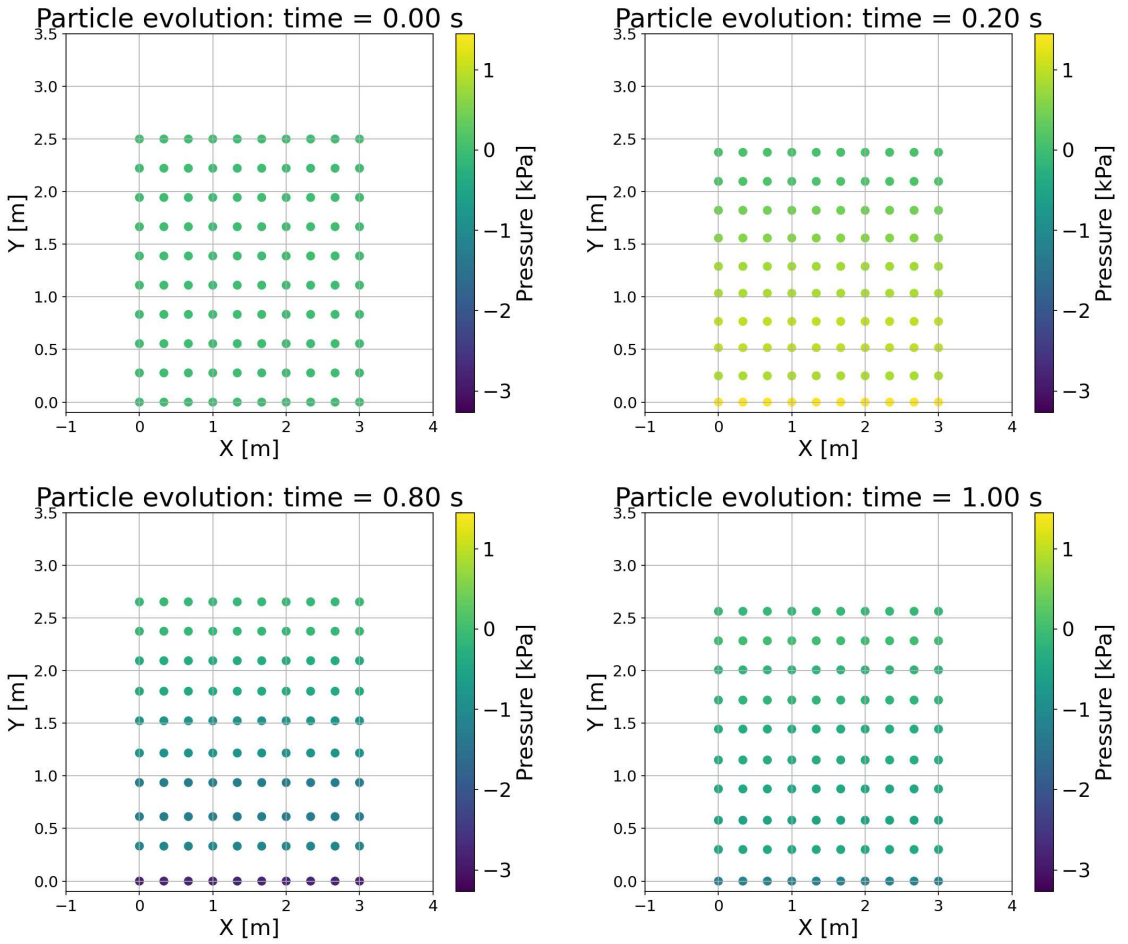


Figure 6.9: Particle position at different time step during the simulation using the implicit time integration scheme. The plots include a color bar visualizing the pressure.

External forces are not considered, but a sinusoidal initial velocity defined by equation (6.27) is applied. Horizontal movements are constrained and particles can move only in vertical direction.
The hyper-elastic material is defined by the coefficient $\gamma = 7$ and the bulk modulus κ which is computed using (6.31) considering $c_{p,0} = 10\ m/s$.
As shown in Figure 6.10, the material exhibits asymmetric oscillatory behavior. In particular, the compressive strength is greater than the tensile strength. This is because the model is nonlinear and does not respond symmetrically under tension and compression. This behavior is also confirmed by the pressure plot in Figure 6.9, which shows higher negative values on pressure.

Moreover, the implicit scheme exhibits an oscillatory response with amplitude constant in time, in both compression and decompression, indicating that the system's energy is conserved. In contrast, the explicit scheme presents signs of instability, compromising the reliability of the results. These effects are a consequence of the nonlinear material employed.

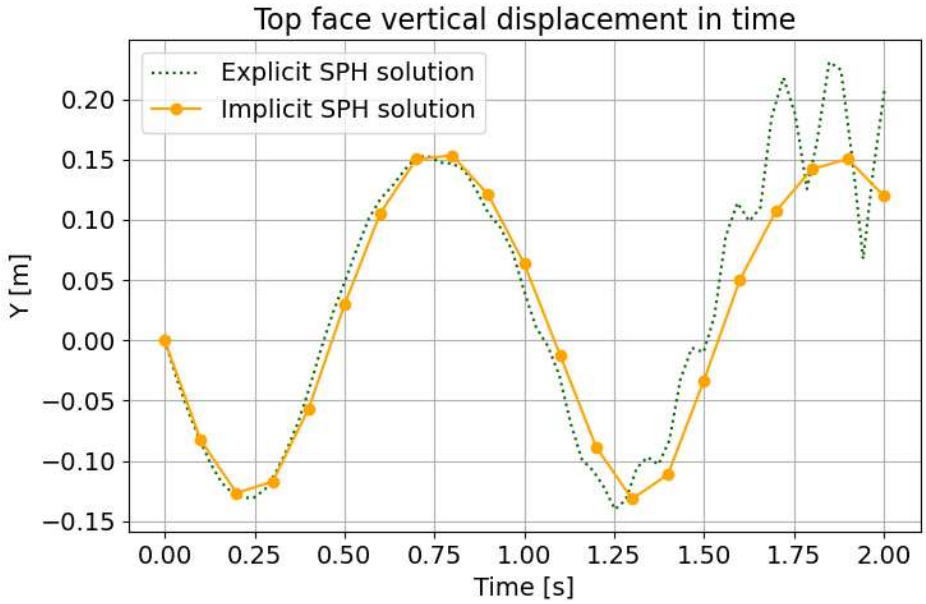
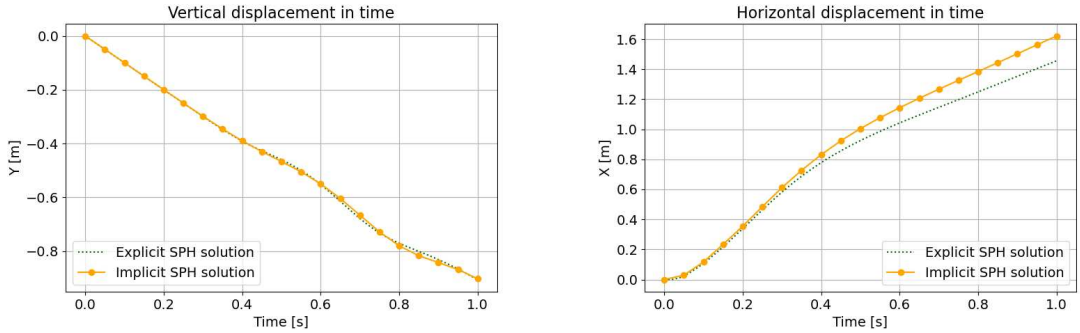


Figure 6.10: Vertical displacement of the top face during the simulation, stabilization terms are not considered. Comparison between implicit and explicit time integration scheme.

Impact test

In this example, the impact of an elastic fluid on a rigid surface is simulated using the new hyper-elastic model. The domain considered is $\Omega = 3 \times 5\ m$, discretized with 225 particles. The particles with y coordinate equal to zero, to simulate the impact, are fixed in vertical direction. Also in this example, the initial motion is given by a constant initial velocity $\boldsymbol{v} = [0, -1]$ (see Figure 6.6).



(a) Vertical displacement of particle at position $\boldsymbol{X} = (1.5,\ 5)$).

(b) Horizontal displacement of particle at position $\boldsymbol{X} = (3,\ 0)$).

Figure 6.11: Comparison between the displacements obtained with implicit and explicit time integration scheme. $\boldsymbol{\mathcal{D}_v}$ is considered while $\mathcal{D}_p = 0$.

The fluid properties are calculated considering $\gamma = 7$ and $c_{p,0} = 10\ m/s$.

The time step is $\Delta t = 0.05\ s$ and its kept constant during the simulation. In Figure 6.11, the displacements in time of two different particles are shown, while in Figure 6.14 the evolution of the full set of particles with the respective pressure values is reported. After the fluid domain is in contact with the wall at time $t = 0$, a compressive shock wave is instantaneously generated (positive pressure) and then followed by a tensile wave (negative pressure). These waves then propagate towards the top free surface and bounce back to the fluid domain. The implicit scheme was adopted to obtain the results shown and the stabilization for the linear momentum equation $\boldsymbol{\mathcal{D}_v} \neq 0$ was used (with $\tau_v = 0.5$). Since a smooth pressure distribution is observed throughout the evolution, there was no need to introduce any pressure stabilization ($\mathcal{D}_p = 0$) into the algorithm.
By comparing this result with the one obtained using the linear elastic model, under identical initial conditions, it is evident that the compressive deformations at the center of the domain are less extreme and less concentrated. This indicates a better behavior under compression of the hyper-elastic model, which contributes to a better stability of the model.
The proposed upwind SPH algorithm performs well without any numerical difficulties.

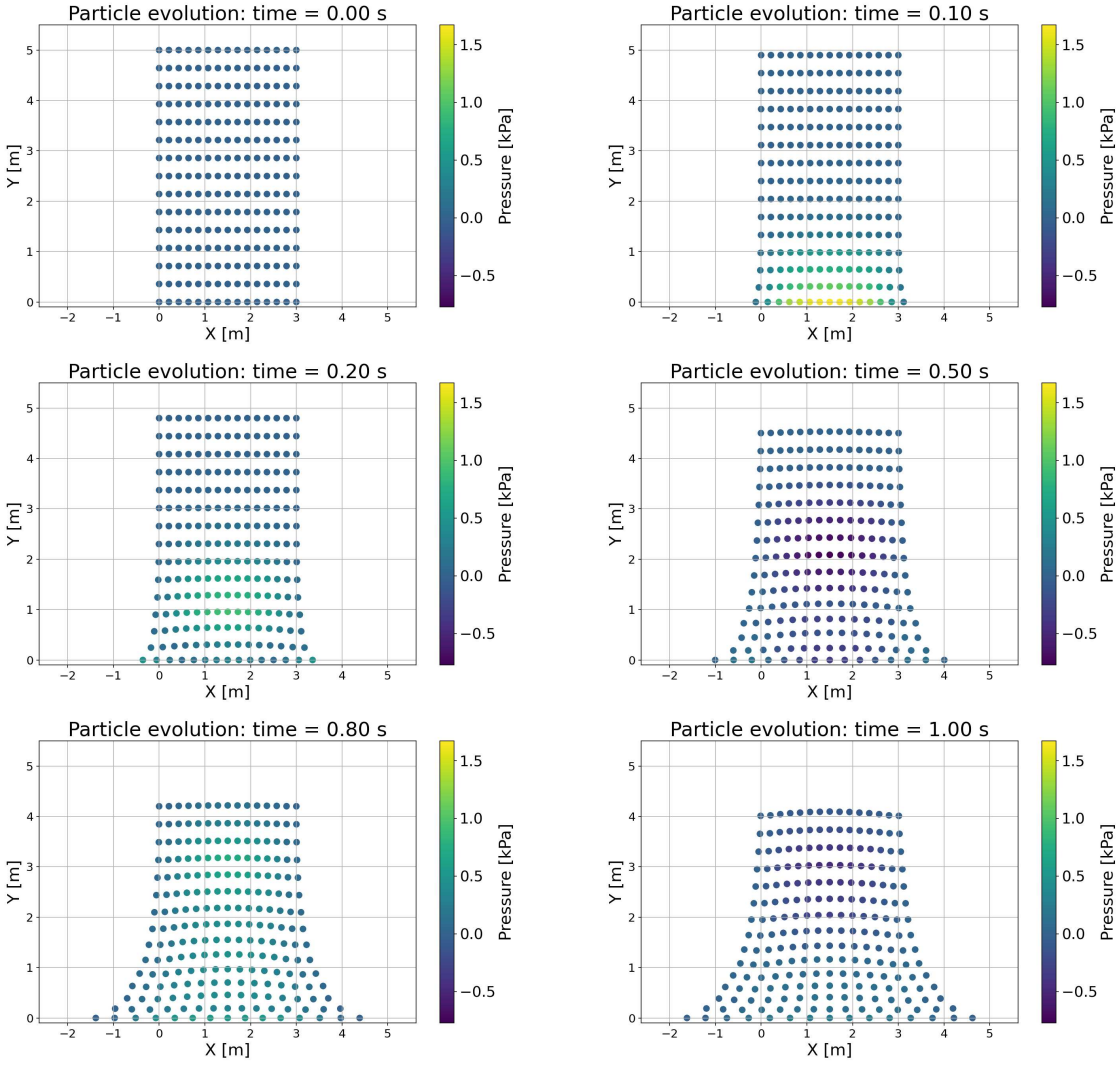


Figure 6.12: Particle evolution during the simulation with implicit time integration scheme. The plots include a color bar visualizing the pressure.

In Figure 6.13 the result with $\boldsymbol{\mathcal{D}_v} = 0$ is reported. In this case, the reciprocal position of the particles, especially at time $t = 0.5\ s$ indicate a significant stretch of the associated volumes, which may lead to instabilities.
Figure 6.14 illustrates the energy loss associated with the dissipation terms. As expected, the energy loss decreases as the discretization becomes more refined.

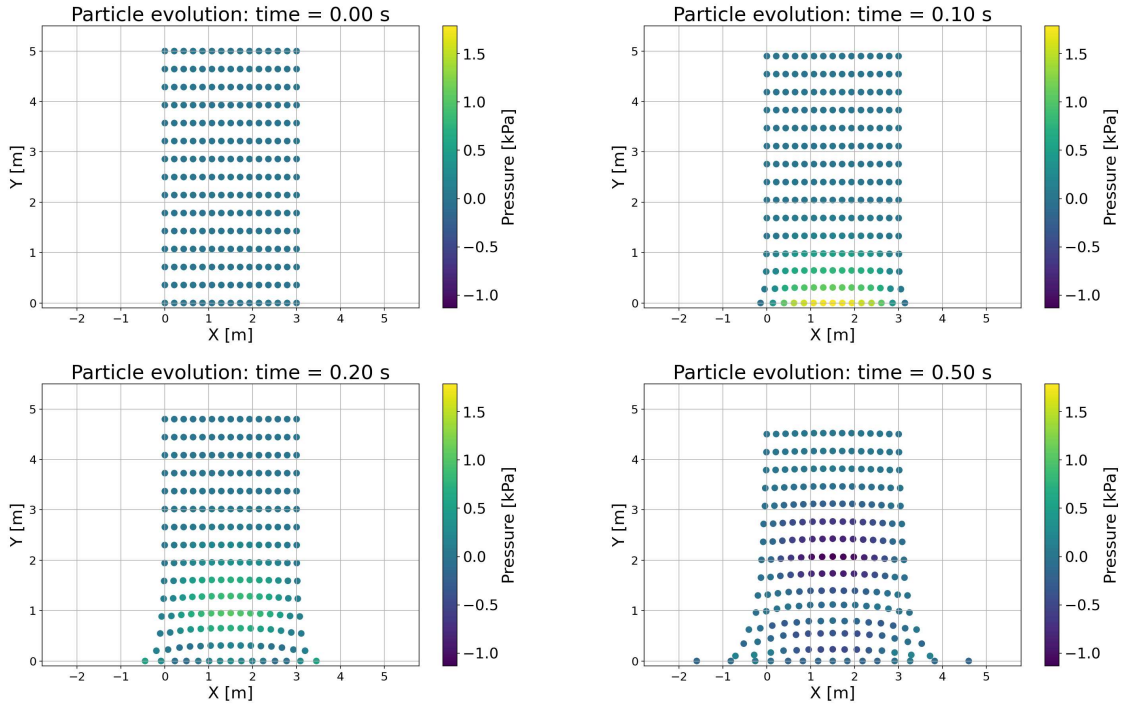


Figure 6.13: Particles evolution during the simulation with implicit time integration scheme, stabilization coefficients τ_v and τ_p are both equal to zero.

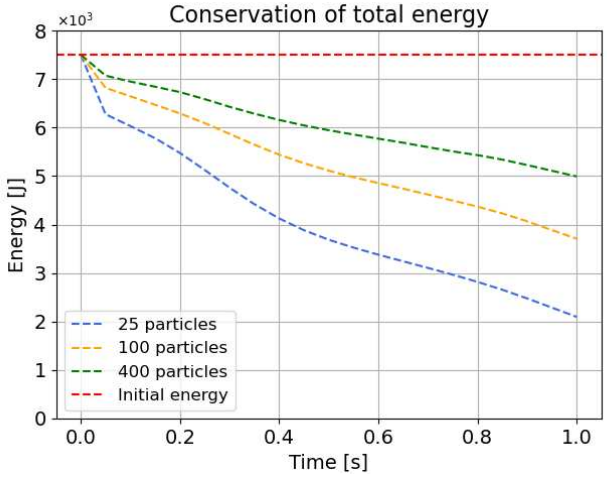


Figure 6.14: Evolution of system's energy comparing different refining discretizations. The value of τ_v is constant and equal to 0.5.

Dam break test

In this section, a dam break problem is presented. This test is performed to see how the novel scheme handles external forces, such as gravity. The domain is a square column

$\Omega = 2.5 \times 2.5\ m$ discretized in 100 particles, in hydrostatic equilibrium confined between a wall on the left and a gate on the right. As anticipated, the material is subject to gravitational acceleration $g = 9.81\ m/s^2$ acting vertically downwards. The fluid properties under consideration are the same as before, density $\rho_0 = 1000\ kg/m^3$, coefficient $\gamma = 7$ and bulk modulus calculated considering $c_{p,0} = 10\ m/s$. In this example, the gate is removed instantaneously and the fluid collapses under gravitational force.

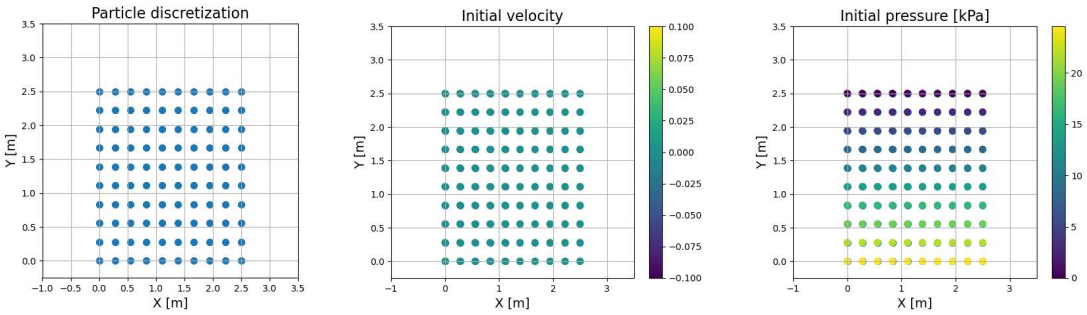


Figure 6.15: Domain discretization and initial conditions

Figure 6.16 and 6.17 compares the evolution of the particles with and without the stabilization term $\boldsymbol{\mathcal{D}_v}$.
Figure 6.17 shows that the absence of stabilization leads to poor quality of the results. In contrast, Figure 6.16, when using stabilization with coefficient $\tau_v = 1$, the particle dynamics align well with the expected behavior based on the prescribed boundaries and initial conditions. Regarding the free surface deformation, the proposed Upwind SPH algorithm produce results comparable to what is reported in [22].

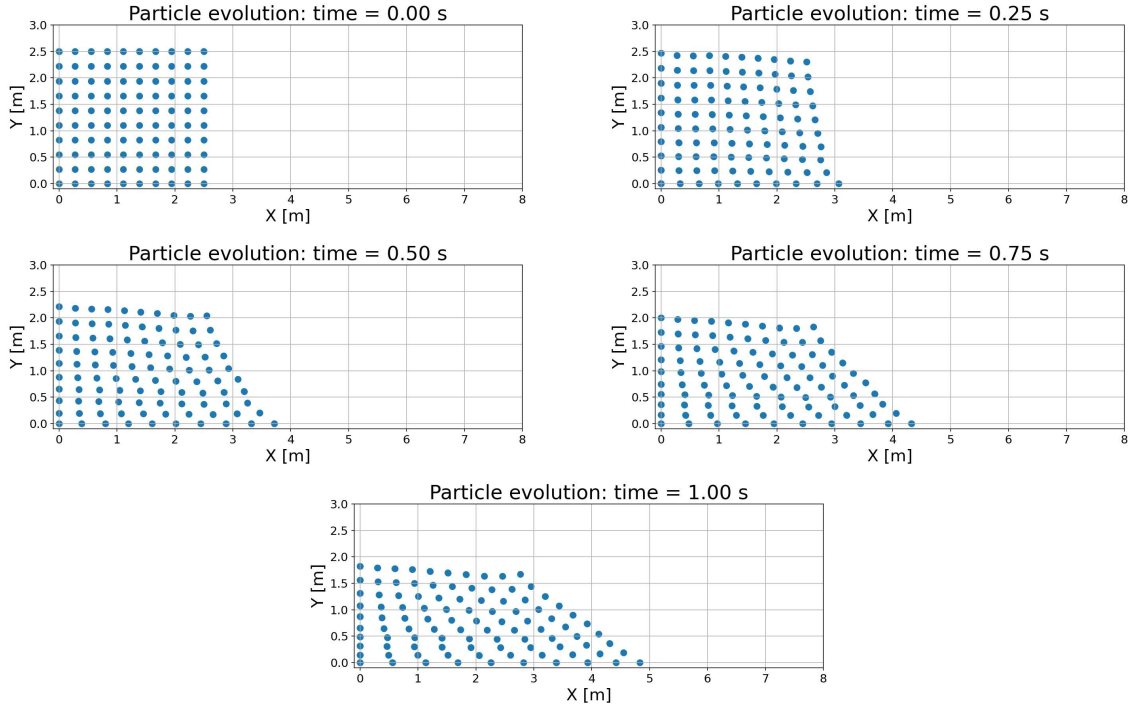


Figure 6.16: Dam break simulation setting $\boldsymbol{\mathcal{D}_v} \neq 0$ with $\tau_v = 1$ and $\mathcal{D}_p = 0$. This figure shows the evolution in time of the position of each particle.

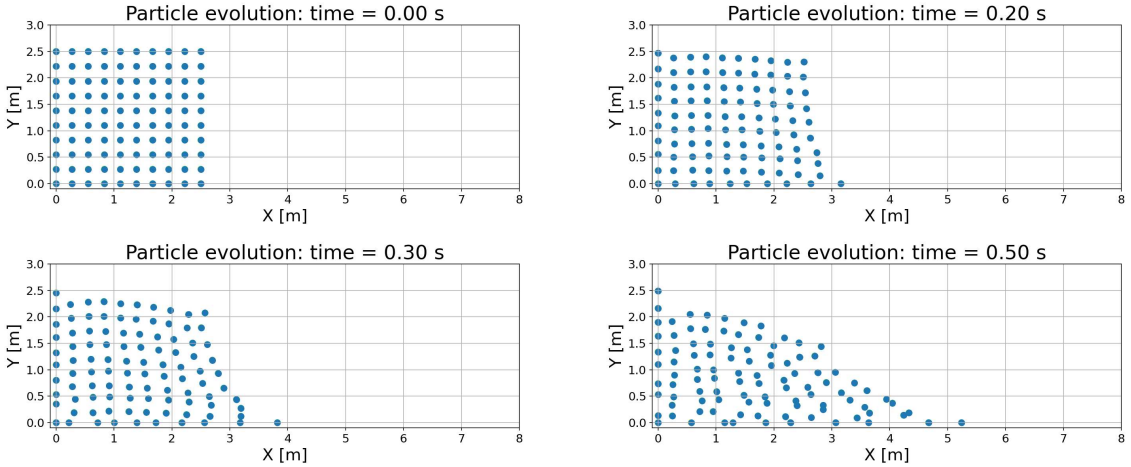


Figure 6.17: Dam break simulation with no stabilization terms considered. This figure shows the evolution in time of the position of each particle.

Chapter 7

Conclusions

This work presents the development of a novel implicit SPH formulation capable of solving a set of benchmark problems both in 1D and in multi-dimension. The main objective was to design and implement an implicit time integration scheme that overcomes the limitations imposed by the restrictive CFL condition. To achieve this, the time derivatives of the system variables are approximated using a second-order Crank–Nicolson integrator; to handle both material and geometric nonlinearities, the Newton-Raphson method was adopted.
In the proposed implicit scheme, the time step Δt does not depend on the discretization or the on material's properties. The scheme has proven to be unconditionally stable, with the only limitation given by the shape and regularity of the solution.
By removing the constraints of explicit solvers, this work enables the study of material evolution over long time spans of time with a limited number of iterations.
Moreover, the use of a linear volumetric elastic law to simulate fluid-like materials is highly limiting because of the material's poor response under compression. On the other hand, adopting an hyper-elastic constitutive model overcomes this issue by better capturing the fluid's behavior under compression.
The stabilization technique employed reduces energy dissipation as the mesh is refined and, at the same time, assure consistency with original problem.
The results presented are limited to 2D simulations. However, the extension to 3D is straightforward. The main challenge lies in the increased complexity when transitioning from 1D to multidimensional problems.

Future developments

A possible future application of this framework could be to develop a numerical model to simulate geodynamical flows, with direct applications to magma dynamics and tectonic plate movements to detect fractures that trigger earthquakes [10]. The magma model coupled with an SPH fracture mechanics module [26], would enable the detection of failure zones that can lead to destructive events.
The SPH-based model developed could have broad applicability in geotechnical engineering, providing a tool to simulate mountain slopes and detect potential avalanches or landslides.
Finally, by addressing fundamental challenges in long-term geodynamical simulations and fracture mechanics, research in this field would have the potential to improve the predictive capabilities of numerical models used in engineering and earth sciences.

Appendix A

Sustainability Analysis and Ethical Implications

Environmental Impact

The environmental impact associated with the development of this thesis has been minimal. The initial phase consisted of review and consultation of academic papers, all conducted digitally without the use of physical resources. The code development was carried out on a personal computer, with energy consumption representing the only notable environmental cost.
The final stage involved running computational simulations that did not require high performance computing infrastructure or significant energy consumption. In fact, the primary objective was to validate the numerical implementation and large scale simulations where not performed.
According to recent data[1], a significant portion of the energy share in Spain is derived from renewable sources. As a result, the carbon footprint associated with the computing activities involved in this research is considered negligible.

Economic Impact

The work required approximately 800 hours of personal computer usage[2]. The laptop uses a 30 W power adapter, so the total energy consumption is estimated as

$$800 \cdot 30 = 24\,000\ Wh = 24\ kWh$$

In Spain, the average electricity price has been around $0.24\,€$ per kWh[3]. Hence, the total energy cost for this work is approximately:

$$24 \cdot 0.24 = 5.76\,€$$

Since the computer was also used for personal tasks, the actual cost attributable to the thesis is likely lower. As the work is purely theoretical and computational, there are no direct environmental risks associated with physical processes. The entire code was developed in `Python`, making use of open-source libraries such as `NumPy`, `SymPy`, and `Matplotlib` for both simulations and visualization of the results. As a result, the development of the thesis had no software related costs.
Although this work has no immediate commercial implications, future realistic adoption could imply significant costs for adaptation to large scale 3D simulations.

[1] `https://www.ree.es/en/statistical-data-of-electricity/annual-report/2024`
[2] MacBook Air (2019 model)
[3] `https://countryeconomy.com/energy-and-environment/electricity-price-household/spain`

Social Impact

The results obtained in this thesis will be accessible to everyone, and the code developed is intended to be integrated in the future into the `KratosMultiphysics`[4] framework.

Ethical Implications

The ethical contribution of this research lies primarily in its potential integration into the open-source framework `KratosMultiphysics`.
By being incorporated into a public and collaborative platform, the developed methodology could be combined with other numerical methods to simulate complex phenomena in various fields of fluid and solid mechanics.
Moreover, the development of an open-source code reflects a commitment to transparency, reproducibility, and accessibility in scientific research. This promotes knowledge sharing, equal access to scientific advancements and collaboration within the research community.

Contribution to the Sustainable Development Goals

This thesis aims to extend the applicability of an existing SPH algorithm to a completely new class of problems. By improving the stability and robustness of the method, the work gives an efficient alternative for the simulation of fluid dynamics in various scenarios. Since the method is general and mesh-free, it can be adapted to various fields, potentially supporting broader sustainability goals. The research aligns with the following UN Sustainable Development Goals[5]:

- **SDG 6 – Clean Water and Sanitation:** The method can be applied to assist in the design and sizing of hydraulic structures, contributing to the management of water resource and risk prevention.
- **SDG 9 – Industry, Innovation and Infrastructure:** The work promotes innovation in numerical methods, contributing to the development of advanced computational tools for engineering and infrastructure design.
- **SDG 13 – Climate Action:** Although indirectly, improved simulation techniques can support understanding and prediction of natural hazards such as avalanches, landslides, and tectonic movements, helping in disaster mitigation strategies.

[4] `https://github.com/KratosMultiphysics/Kratos`
[5] `https://sdgs.un.org/goals`

Bibliography

[1] `https://interactivecomputergraphics.github.io/physics-simulation/examples/sph_kernel.html`.

[2] Miquel Aguirre, Antonio J. Gil, Javier Bonet, and Chun Hean Lee. An upwind vertex centred finite volume solver for lagrangian solid dynamics. *Journal of Computational Physics*, 300:387–422, 2015.

[3] J. Bonet and T.-S.L. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180, 1999.

[4] Javier Bonet and Sivakumar Kulasegaram. Remarks on tension instability of eulerian and lagrangian corrected smooth particle hydrodynamics (csph) methods. *International Journal for Numerical Methods in Engineering*, 52(11):1203–1220, 2001.

[5] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[6] Eleni Chatzi, Giuseppe Abbiati, and Konstantinos Agathos. The finite element method for the analysis of non-linear and dynamic systems: Non-linear dynamics part i, 2017.

[7] Richard Courant, Kurt Friedrichs, and Hans Lewy. On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, 11(2):215–234, 1967.

[8] Carl T Dyka and Robert P Ingel. An approach for tension instability in smoothed particle hydrodynamics (sph). *Computers & structures*, 57(4):573–580, 1995.

[9] Sandra Forte, Luigi Preziosi, and Maurizio Vianello. Corpi e deformazioni. *Meccanica dei Continui*, pages 1–61, 2019.

[10] Taras Gerya. *Introduction to numerical geodynamic modelling*. Cambridge University Press, 2019.

[11] AJ Gil, Z Zhang, O Hassan, and K Morgan. Parallel multigrid detached eddy simulation algorithm for three-dimensional unsteady incompressible flows on unstructured meshes. *Journal of aerospace engineering*, 19(4):271–280, 2006.

[12] Antonio J Gil, Chun Hean Lee, Javier Bonet, and Miquel Aguirre. A stabilised petrov–galerkin formulation for linear tetrahedral elements in compressible, nearly incompressible and truly incompressible fast dynamics. *Computer Methods in Applied Mechanics and Engineering*, 276:659–690, 2014.

[13] Morton E Gurtin, Eliot Fried, and Lallit Anand. *The mechanics and thermodynamics of continua*. Cambridge university press, 2010.

[14] Jibran Haider, Chun Hean Lee, Antonio J Gil, and Javier Bonet. A first-order hyperbolic framework for large strain computational solid dynamics: an upwind cell centred total lagrangian scheme. *International Journal for Numerical Methods in Engineering*, 109(3):407–456, 2017.

[15] Osama I Hassan, Ataollah Ghavamian, Chun Hean Lee, Antonio J Gil, Javier Bonet, and Ferdinando Auricchio. An upwind vertex centred finite volume algorithm for nearly and truly incompressible explicit fast solid dynamic applications: Total and updated lagrangian formulations. *Journal of Computational Physics: X*, 3:100025, 2019.

[16] Gordon R Johnson, Robert A Stryk, and Stephen R Beissel. Sph for high velocity impact computations. *Computer methods in applied mechanics and engineering*, 139(1-4):347–373, 1996.

[17] Chun Hean Lee, Antonio J. Gil, and Javier Bonet. Total lagrangian sph algorithm: First order conservation law for fast solid dynamics. MATLAB code.

[18] Chun Hean Lee, Antonio J Gil, and Javier Bonet. Development of a cell centred upwind finite volume algorithm for a new conservation law formulation in structural dynamics. *Computers & Structures*, 118:13–38, 2013.

[19] Chun Hean Lee, Antonio J Gil, Ataollah Ghavamian, and Javier Bonet. A total lagrangian upwind smooth particle hydrodynamics algorithm for large strain explicit solid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 344:209–250, 2019.

[20] Chun Hean Lee, Antonio J Gil, Giorgio Greto, Sivakumar Kulasegaram, and Javier Bonet. A new jameson–schmidt–turkel smooth particle hydrodynamics algorithm for large strain explicit fast dynamics. *Computer Methods in Applied Mechanics and Engineering*, 311:71–111, 2016.

[21] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[22] Kenny WQ Low, Chun Hean Lee, Antonio J Gil, Jibran Haider, and Javier Bonet. A parameter-free total lagrangian smooth particle hydrodynamics algorithm applied to problems with free surfaces. *Computational Particle Mechanics*, 8, 2021.

[23] Joe J Monaghan. Smoothed particle hydrodynamics. *In: Annual review of astronomy and astrophysics. Vol. 30 (A93-25826 09-90), p. 543-574.*, 30:543–574, 1992.

[24] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[25] Nathan M Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[26] Mohammad Naqib Rahimi and Georgios Moutsanidis. A smoothed particle hydrodynamics approach for phase field modeling of brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 398:115191, 2022.

[27] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[28] Riccardo Rossi. *Light weight Structures: Structural Analysis and Coupling Issues*. Phd thesis, Università di Bologna, 2005.

[29] SymPy Developement Team. Sympy: Python library for symbolic mathematics, 2025.

[30] Damien Violeau and Benedict D Rogers. Smoothed particle hydrodynamics (sph) for free-surface flows: past, present and future. *Journal of Hydraulic Research*, 54(1):1–26, 2016.