Title

Delta Voronoi Smoothed Particle Hydrodynamics, δ-VSPH

Author

Fernandez Gutierrez, David

Publication Date

2018

Delta Voronoi Smoothed Particle Hydrodynamics, δ-VSPH

By

David Fernández Gutiérrez

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Tarek. I. Zohdi, *Chair*
Professor Ömer Savaş
Professor Mohammad-Reza Alam
Professor Khalid M. Mosalam

Spring 2018

Delta Voronoi Smoothed Particle Hydrodynamics, δ-VSPH

by

David Fernández Gutiérrez,

UC Berkeley, Mechanical Engineering Department.

Abstract

Delta Voronoi Smoothed Particle Hydrodynamics, δ-VSPH

by

David Fernández Gutiérrez

Doctor of Philosophy in Mechanical Engineering

University of California, Berkeley

Professor Tarek I. Zohdi, *Chair*

A Lagrangian scheme that combines Voronoi diagrams with smoothed particle hydrodynamics (SPH) for incompressible flows has been developed. Within the Voronoi tessellation, the differential operators are discretized by adding contributions from neighboring cells following the Voronoi particle hydrodynamics (VPH) method, (Serrano and Español, *Phys. Rev. E*, 64, 2001; Hess and Springel, *Month. Not. of the Royal Astronomical Society*, 406(4), 2010). This approach is structurally similar to the one used by SPH, although the neighbor set is larger in the latter. Due to this similitude, and the feasibility to approximate the fluid as weakly compressible in both SPH and VPH, coupling the two methods is a natural choice explored in detail in this thesis.

The Voronoi cells allow enforcing the boundary conditions directly on the portion of their perimeter in contact with the boundary. Two sub-domains are then defined based on the proximity to the boundaries. The VPH formulation is used in those particles close to solid boundaries, where SPH consistency and boundary conditions implementation become problematic. The SPH formulation is limited to the region such that no boundary falls within the kernel domain. Some overlapping of both sub-domains is allowed to provide a buffer zone to progressively transition from one method to the other. In this zone, fields are calculated using both methods and then coupled by a weighting function that depends on the distance to the closest boundary. As particles move, they can cross the buffer zone and change sub-domains.

The unbounded nature of Voronoi tessellations requires further treatment close to boundaries and free-surfaces to ensure that the Voronoi cells associated with the particles do not extend beyond the fluid domain. Boundaries are discretized as segments, or D-dimensional triangles in higher dimensions, and virtual particles are added mirroring the particles next to the boundaries. Since the boundary becomes a symmetry plane, no Voronoi cell can cross it. However, Voronoi cells resulting from the tessellation are always convex, so an additional trimming algorithm has been implemented to deal with concave boundaries. In the case of free-surfaces, a new detection algorithm determines if a particle is next to the free-surface and estimates the free-surface normal direction, which is used to bound the corresponding Voronoi cell. Periodic and inlet/outlet boundary conditions are also included in the coupled scheme.

The δ-SPH correction proposed by Cercos-Pita et al., *Applied Math. Modelling*, 40(19), 2016, is extended to the VPH formulation. In addition, the density field is reinitialized every certain number of time steps following the scheme proposed by Colagrossi and Landrini, *J. Comp. Phys*, 191, 2003. A shifting algorithm inspired by Lloyd, *IEEE Trans. of Inform. Theory*, 28(2), 1982, is also included to avoid excessive deformation of the Voronoi cells.

The code key subroutines are written in C++, and the main time-stepping program is implemented in Matlab. A leap-frog numerical scheme is used to iterate forward in time. The program architecture is summarized graphically, with a detailed analysis of the neighbor boundaries and particles search algorithm included afterwards. A linear damping term is used during the initialization process to mitigate possible inconsistencies from the user-defined initial conditions. Finally, the energy balance of the coupled scheme is discussed.

The method presented in this thesis is suitable for free-surface flows with moving solid boundaries within the fluid domain. However, prior to extending it to such complicate problems, the accuracy of the coupled scheme is discussed by means of a set of well-known verification benchmarks.

Keywords: CFD – SPH – Voronoi – Coupling

To my wife, Rachael, and to my mother, Sara,
you are always a beacon that helps me navigate through this life

CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGEMENTS

It feels as yesterday when I was at the Towing Tank of the Technical University of Madrid and heard for the first time the word “SPH”. Many twists after, I finally could learn and contribute to this field. It has been a privilege to be able to cross it out of my bucket list.

I would like first to thank my advisor, Prof. Tarek I. Zohdi, for giving me a home when I needed it and the freedom to unleash my curiosity. I truly appreciate your guidance and support to finish this thesis. I would like also to thank the members of my committee, Prof. Ömer Savaş, Prof. Mohammad-Reza Alam, and Prof. Khalid M. Mosalam, for all your comments to help me improve my work. I owe a special recognition to Prof. Antonio Souto Iglesias, you have always been an invaluable source of inspiration and knowledge.

Nothing of what I have accomplished could have been possible without the continuous example, encouragement, and support of my family, I cannot thank you enough. There is one person here that shines with a special light, Rachael, you are the heart and soul of this PhD.

My gratitude also to all the instructors that chose me as a Graduate Student Instructor for ME163 and ME107, it has been an honor to learn from you. I would like to extend my acknowledgement to the Rafael del Pino Foundation and the Graduate Division for the grants received to support my studies at UC Berkeley.

Last but not least, I would like to acknowledge all the students, professors, and friends who have made me a better engineer and, above all, a better person. A special place in my heart is reserved to my partner in crime, Abdulrahman Jbaily, and to my fellow grad students with whom I have shared this battleground that we call Berkeley: Claire, Lu, Spencer... the list will be too long to name everybody. I must thank as well all my Spanish friends for bearing with me despite the distance, you always make me feel at home! Together all, we will keep making the world smaller. Those that know me are well aware that soccer runs through my veins, so I couldn’t finish without thanking all my MEAT soccer buddies for letting me play in the fields.

Thank you everybody for making these past years unforgettable, you know all who you are.

”Make everything as simple as possible, but not simpler”
- Albert Einstein

”The particle method is not only an approximation of the continuum fluid equations, but also gives the rigorous equations for a particle system which approximates the molecular system underlying, and more fundamental than the continuum equations”
- Joe J. Monaghan (paraphrasing J. von Neumann)

NOMENCLATURE

VORONOI DIAGRAM COMPONENTS

cell	Region assigned to each seed such that it represents the portion of the domain closer to the linked seed than to any other
edge	Connection between two vertices of a face
face	$D - 1$ simplex of a D-dimensional tessellation, representing the contact zone between two neighboring seeds
perimeter	Set of faces that enclose a cell
seeds	Points (particles or virtual points) that originate the tessellation
tessellation	Set of cells that represent a complete partition of the D-dimensional space
vertex	Node of the tessellation common to multiple faces

ACRONYMS

ALE	Arbritray Lagrangian–Eulerian
AV	Artificial Viscosity
BC	Boundary condition
BEM	Boundary Element methods
BVP	Boundary virtual point
CFD	Computational Fluid Dynamics
DPD	Dissipative Particle Dynamics
DSMC	Direct Simulation Monte Carlo
FS	Free-surface
FSVP	Free-surface virtual point
GPU	Graphics processing unit
HPC	High-performance computing
I/O	Inlet/Outlet
LBM	Lattice Boltzmann methods
LES	Large eddy simulations
LHS	Left hand side of an equation
LS	Level Set
MD	Molecular dynamics
MLS	Moving least squares
MPM	Material point methods
NS	Navier–Stokes
ODE	Ordinary differential equation
PDE	Partial differential equation
PIC	Particle-In-Cell
RHS	Right hand side of an equation

SPH	Smoothed Particle Hydrodynamics
VM	Vortex methods
VOF	Volume of Fluid
VPH	Voronoi Particle Hydrodynamics
WC2	Second-order continuous Wendland kernel

Variables (*Roman characters*)

$\mathbf{0}$	Zero vector
$\mathbf{1}$	Arbitrary constant unit vector
$\mathcal{AR}$	Aspect ratio
A	Area
a	Arbitrary fluid particle
b	Neighboring particles of a given particle a
c	Speed of the sound
$\boldsymbol{c}_{ab}$	vector going from the midpoint between particles a and b to the centroid of their associated Voronoi cell face
C_B	Set of boundaries linked to each grid cell
C_F	Set of Voronoi faces linked to each Voronoi cell
CFL	Courant–Friedrichs–Lewy coefficient
$C\mathcal{L}_P$	Particle indexes arranged sequentially by the grid cell that they belong to
$C\mathcal{L}_C$	Position of the first particle in $C\mathcal{L}_P$ for each grid cell
$\mathbb{D}$	Rate of deformation tensor
D	Number of dimensions of the fluid spatial domain
d	Distance to the closest boundary
d_{max}	Distance from a particle to its furthest Voronoi cell face
d_g	Distance from a particle to the centroid of its Voronoi cell
$\mathcal{E}$	Energy
$\mathcal{E}_C$	Elastic energy
$\mathcal{E}_I$	Internal energy
$\mathcal{E}_K$	Kinetic energy
$\mathcal{E}_M$	Mechanical energy
$\mathcal{E}_{\text{num}}$	Numerical dissipated energy
$\mathcal{E}_P$	Potential energy
$\mathcal{E}_T$	Thermal energy
$\mathcal{E}_{\text{tot}}$	Total energy
$\boldsymbol{e}$	Unit vector
e	Internal energy per unit mass
$\boldsymbol{F}$	Force
F	Normalized kernel derivative

$\boldsymbol{f}$	Body forces per unit mass, or arbitrary vector field
f	Arbitrary scalar field
$\mathcal{F}_P$	Set of particles linked to each Voronoi face
$\mathcal{F}_V$	Set of Voronoi nodes linked to each Voronoi face
Fr	Froude number
$\boldsymbol{g}$	Gravity acceleration vector
g	Gravity acceleration magnitude
h	Smoothing length
$\mathbb{I}$	Identity tensor
$\mathcal{K}_{cc}$	Set local relative indexes of neigboring grid cells
$\boldsymbol{k}_c$	Local grid cell index vector (gathers the local indexes in each dimension)
k_c	Global grid cell index
$\boldsymbol{\Delta k}_{cc}$	Relative global indexes of neighboring grid cells
k_k	Kernel normalization constant (depends on D)
l	Length
M	Normalized kernel function
m	Mass
Ma	Mach number
N, N_p	Number of particles
$\boldsymbol{N}_c$	Vector containing the number of grid cells in each dimension
N_c	Number of grid cells
$\boldsymbol{n}$	Normal vector
n	Time step index
Δn_ρ	Density re-initialization frequency
p	Pressure
$\mathbb{O}$	Zero tensor
$\mathcal{O}$	Order of magnitude
$\mathcal{P}$	Power
$\mathcal{P}_{\text{num}}$	Numerical dissipation rate
$\mathcal{P}_V$	Viscous dissipation rate
$\dot{\boldsymbol{Q}}$	Rate of heat added to a system
$\dot{Q}$	volumetric heat deposition
$\boldsymbol{q}$	heat flux vector
q	Normalized distance to a particle ($q = \\|\boldsymbol{r} - \boldsymbol{r}_a\\|/h$), or normalized location across the buffer ($q = (d - \kappa h_{\text{max}})/l_{\text{buffer}}$)
Re	Reynolds number
$\boldsymbol{r}$	Position vector
$\boldsymbol{\mathcal{S}}$	Surface vector, oriented normal to it and with magnitude $\mathcal{S}$
$\mathcal{S}$	Surface magnitude

t	Time
Δt	Time step increment
t_s	Initial stabilization time
$\boldsymbol{u}$	Fluid velocity vector
u	Velocity magnitude
u^{sig}	Inter-particle signal speed
$\mathcal{V}$	Volume
$\dot{\mathcal{W}}$	Rate of work done by a system
W	Kernel function
$\mathcal{X}$	Set of point coordinates
Δx	Particle size
$\boldsymbol{x}$	Position vector of a Voronoi diagram node
$\boldsymbol{x}_g$	Position vector of a Voronoi cell centroid

VARIABLES (*Greek symbols*)

α	Artificial viscosity coefficient
β	MLS coefficients
γ	Coefficient that defines the level of compressibility of the fluid
δ	Diffusive correction term coefficient or Dirac delta function
ε	Normalized buffer thickness
ζ	Allowed deviation threshold around cell centroid (shifting algorithm coefficient)
η	Relative density variation
κ	Coefficient that defines the compact support radius of the kernel (multiple of h)
λ	Second coefficient of dynamic viscosity
μ	First coefficient of dynamic viscosity
ν	Kinematic viscosity ($\nu = \mu/\rho$)
ξ	Linear damping intensity coefficient, usually defined in terms of $\xi\,\Delta t$
ρ	Density evolved using the continuity equation
$\check{\rho}$	Density computed from the particle distribution
$\underline{\underline{\sigma}}$	Stress tensor
$\underline{\underline{\tau}}$	Viscous stress tensor
Ω	Fluid domain
$\partial\Omega$	Fluid domain boundary
ω	Coupling weight

SUPERSCRIPTS

$*$	Predicted value or local integration variable
$\star$	Kinetic energy evolved in time, including only the artificial viscosity numerical dissipation component
0	Initial state ($n = 0$)

α, AV	Artificial viscosity component
B	Boundary virtual point
FS	Free-surface virtual point
IO	Inlet/outlet particle
P	Periodic particle
p	Pressure component
T	Tensor transpose
τ, V	Viscous component

SUBSCRIPTS

$\perp$	Orthogonal
0	Reference values
B	Associated with a solid boundary
FS	Associated with free-surface
IO	Associated with an inlet/outlet boundary
P	Associated with a periodic boundary

MATHEMATICAL OPERATORS

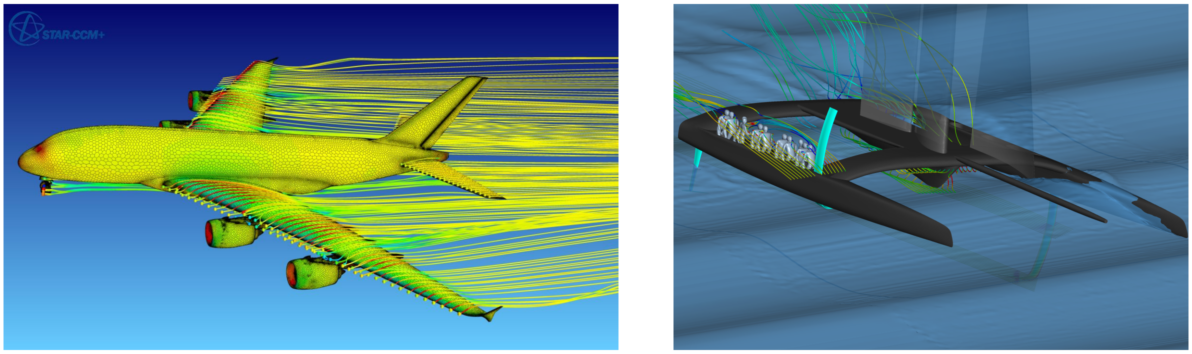
$\wedge$	Exterior product
$\cdot$	Dot product
$:$	Double dot product
$\|\| \ \|\|$	Euclidean 2-norm
$[\ ]$	Volume averaged value
$<\ >$	Smoothed value
$\frac{\mathrm{d}}{\mathrm{d}t}$	Lagrangian time derivative
∇	Gradient differential operator (*grad*)
$\nabla\cdot$	Divergence differential operator (*div*)
∇^2	Laplacian differential operator

CHAPTER 1

INTRODUCTION

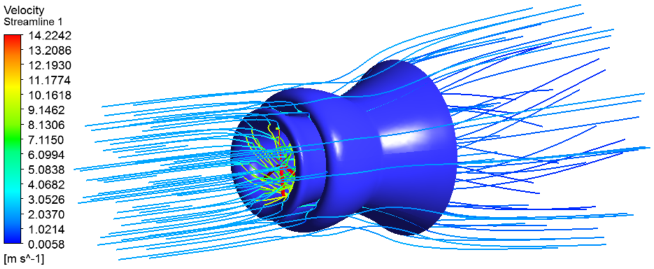
1.1 OVERVIEW

The situations where it would be extremely useful to model accurately fluid flows are innumerable. Despite the fact that the Navier-Stokes equations that govern the motion of fluids were formulated in the early 19$^{\text{th}}$ century, few exact solutions are known (Drazin and Riley, 2006). Practical applications usually rely on computational fluid dynamics (CFD), which provide numerical solutions under different assumptions based on the nature of the flow. Some examples are shown in Fig. 1.1.



(a) Simulation on an Airbus A380, showing pressure field on the structure and streamlines from the nose wheel and the wings leading edges, `https://mdx2.plm.automation.siemens.com/cfdImage/simulation-airbus-a380`

(b) Simulation of an America's Cup AC72 catamaran under regular waves, `http://www.caponnetto-hueber.com`



(c) Simulation of the flow through a marine current turbine (Fernández-Gutiérrez et al., 2014)

Figure 1.1: CFD simulations

The equations of motion can be formulated in an Eulerian or a Lagrangian form. The Eulerian approach is typical for fixed-grid solvers, while the Lagrangian one is suitable for meshfree particle methods since they follow a fluid parcel instead of a grid node. The existence of a grid allows using higher order numerical operators, but requires modifying the mesh when the shape of the fluid domain changes. *Arbitrary Lagrangian–Eulerian (ALE)* methods deal with this type of problems, but become significantly more complex and computationally expensive (Hirt et al., 1997). Particle methods avoid this burden, though they rely on lower-order operators.

Many of the physical phenomena of interest involve free-surfaces, which exist in multi-phase flows with large density differences between the phases. Potential flow solvers such as *Boundary Element Methods (BEM)* can be used with low-deformed free-surfaces (Yeung, 1982). However, more advanced solvers are required when this is not the case and surface-breaking, fragmentation, and phase entrainment occur. Eulerian methods using techniques such as *Volume of Fluid (VOF)* or *Level Set (LS)* are common alternatives (Hirt and Nichols, 1981; Sethian, 1999). Nevertheless, particle methods become very attractive for this type of problems because the free-surface boundary condition is intrinsically solved in their formulation, not requiring any special treatment. *Smoothed particle hydrodynamics (SPH)* solvers are among the most popular ones in this category (Monaghan, 2012; Springel, 2010b) .

The present thesis focuses on the SPH method, and in particular on how it models the interaction of the fluid with solid boundaries. An alternative approach is proposed, which takes the particles close to the boundaries as seeds to partition the spatial domain into Voronoi cells (Fig 1.2). Each cell represents the space closer to one particle than to any other (Voronoi, 1908), being a logical choice to link non-overlapping zones to each particle. Moreover, instead of smoothing, an alternative formulation is employed to approximate the spatial differential operators based on the Voronoi tessellation. This approach allows us to enforce directly the boundary conditions on those cells in contact with the boundary, removing the requirement for fictitious repulsive boundary or ghost particles as it is usually done in standard SPH solvers.

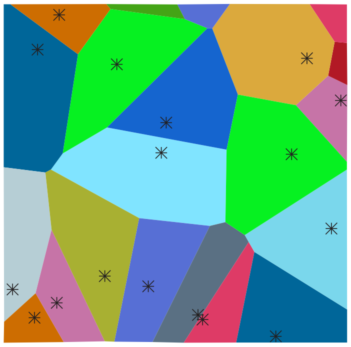


Figure 1.2: Portion of a 2D Voronoi diagram, with each cell identified with a different color

1.2 MOTIVATION

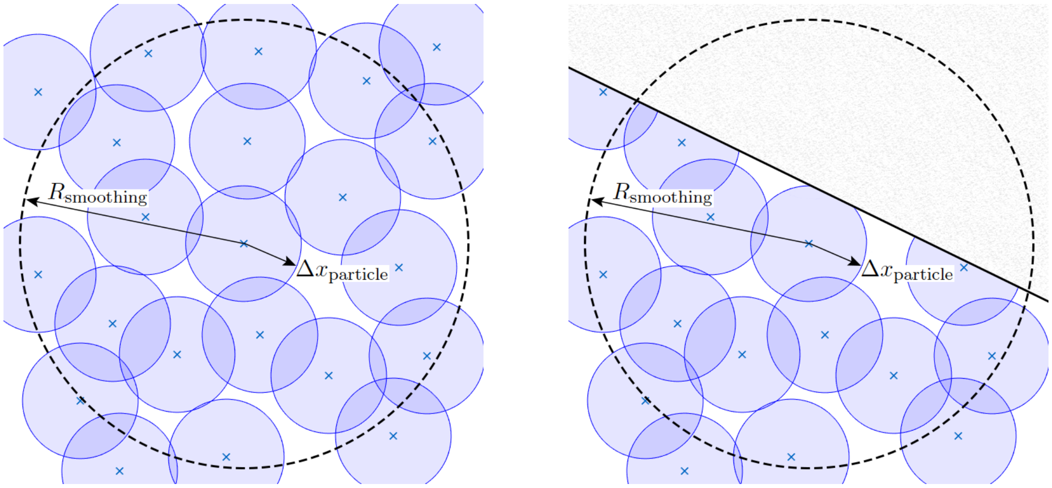
The nature of fluid flows can be very different, being usually characterized by non-dimensional parameters such as Reynolds number, Re, Mach number, Ma, Froude number, Fr, etc. Particular flow conditions allow us to apply certain simplifications to tackle the governing equations and obtain approximate solutions.

Logically, each numerical method is adequate for a certain type of flows. However, there are many situations where we can identify various fluid regimes within the computational domain. Ideally, we would like to combine multiple methods, applying each where is more suitable. This task is usually very challenging due to incompatibilities between the methods, but it expands the applicability of the numerical scheme significantly when achieved.

The *smoothed particle hydrodynamics* method is known for its flexibility to easily model complex physics and its outstanding conservation properties, but also for its limitations in discretization, adaptivity, accuracy and implementation of boundary conditions, among others. The main motivation of this thesis is precisely to use the "multi-method" approach to tackle these limitations instead of performing modifications to the SPH scheme, which in any case penalizes the conservation properties.

The meshless nature of SPH makes it suitable for simulating moving objects immersed in a fluid, such as swimming bodies, propellers, or flapping sails for example. It is even more applicable when these bodies interact with free-surfaces due to the ease to incorporate them in the numerical scheme, as explained before. The flows around energy harvesting offshore structures, like coastal marine current turbines, are good examples for such conditions. The correct implementation of solid boundary conditions is critical in these applications to model the fluid-structure interaction. Consequently, it was chosen as the area of SPH to focus on.

In SPH, we define a set of interpolation nodes over the domain, named particles, and assign to each a certain mass m_i. Based on the density field $\rho(\boldsymbol{r}, t)$, we estimate at a given time $t^{(n)}$ the volume[1] associated with each particle as $\mathcal{V}_i = m_i \,/\, \rho(\boldsymbol{r}_i, t^{(n)})$. The value of any field variable at a particular location can then be approximated, namely smoothed, based on the particles around it within certain radius. However, because the particles are randomly placed, the total computed volume from these neighboring particles does not match the actual fluid volume within that radius (Fig. 1.3(a)). As seen later in Sec. 3.1, this fact has an important impact on the consistency of the smoothing. Moreover, the deviation between volumes is even larger near the boundaries since part of the smoothing domain falls beyond the boundary (Fig. 1.3(b)).



(a) Fully immersed in the fluid

(b) Close to boundaries

Figure 1.3: Overlapping and voids in SPH volume discretization, $R_{\text{smoothing}} \approx 4\Delta x_{\text{particle}}$.

---

[1] Note that the term "volume" represents the portion of the fluid domain associated with the particle. In 2D problems, it will be a surface.

In order to avoid this issue, an alternative method that provides a consistent partition of the domain using Voronoi diagrams was envisioned (Fig. 1.4). Furthermore, Sec. 3.2 shows how we can develop a formulation similar to SPH based on the information from the Voronoi diagram. Instead of relying on all particles within certain distance, the spatial differential operators are computed based on the faces that delimit each cell and the field values of the neighbor/boundaries to which they correspond. No dependence on the volume mismatch exists in this case, achieving our goal.

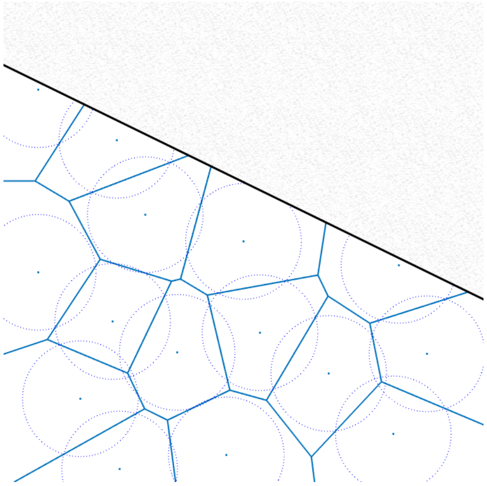


Figure 1.4: Voronoi diagram with the same particle distribution as in Fig. 1.3(b). Continuous lines delimit Voronoi cells, while dotted lines indicate the SPH particle radius

The present work establishes the basis of a coupling between the SPH and Voronoi formulations, so it can be applied at large scale in the future. Long term applications include using the Voronoi scheme to transition from SPH to fixed-grid methods, where Lagrangian formulations can be employed close to moving objects and Eulerian ones in the far field. This approach can also be useful for problems dealing with solidifying materials, common in processes like additive manufacturing.

1.3 LITERATURE REVIEW

1.3.1 PARTICLE METHODS

It is very intuitive to think of a fluid as a set of moving elements. However, the use of particle methods is relatively recent (Li and Liu, 2004), despite their mentioned advantages in dealing with free-surfaces and moving-boundaries. Arguably, this can be attributed to larger computational requirements, mathematical formulations that lay further from analytic solutions, etc. In any case, significant advancements have been achieved in particle methods in the last decades[2], which remain as an active research area in the fluid dynamics community (Oñate and Owen, 2011). Looking at the fluid itself, we can recognize three levels of abstraction:

---

[2] `https://en.wikipedia.org/wiki/Meshfree_methods`

- Molecular level. *Molecular dynamics (MD)* simulations reach this level (Rapaport, 1996; Allen and Tildesley, 2017). They are useful to determine thermo-mechanical material properties, and are mainly used in chemistry physics, material science, molecular biology, and nano-engineering. At larger scales, they become unfeasible due to the number of elements.

- Mesoscale level, considered as intermediate scale that links the molecular behavior to the continuum models used at the macroscale level. Methods that focus on this level are for example *Direct Simulation Monte Carlo (DSMC)* (Bird, 1976) or *Dissipative Particle Dynamics (DPD)* (Español and Warren, 1995, 2017). *Lattice Boltzmann methods (LBM)* also fall within this category, and are becoming very popular recently thanks to their ease to implement and parallelize (Chen and Doolen, 1998). These methods are capable of solving the flow through porous media intrinsically, although they have difficulties under high *Ma* or when free-surfaces are present.

- Macroscale level. The fluid is treated as a continuum medium at this level, with the numerical schemes able to tackle problems ranging from micrometers to millions of kilometers. Particles in this case conform a set of nodes scattered through the fluid domain that are tracked in a Lagrangian way. *Smoothed particle hydrodynamics* method belongs to this group, which evolved from the *Particle-In-Cell (PIC)* method developed by Evans and Harlow (1957). Among other common alternatives, *Vortex methods (VM)* (Chorin, 1973; Chorin and Bernard, 1973; Leonard, 1980; Wang, 2016; Wang and Yeung, 2016a,b) can be used for incompressible fluids, and more recently *Material point methods (MPM)* (Sulsky et al., 1994) are receiving a growing interest for granular flows.

1.3.2 SMOOTHED PARTICLE HYDRODYNAMICS

This thesis focuses on SPH for its great flexibility and potential to tackle complex problems found in engineering, as discussed before. The SPH method was originally presented by Lucy (1977) and Gingold and Monaghan (1977) for astro-dynamical applications. Most of the work done in the following years was done by Prof. Monaghan and his collaborators (Monaghan and Gingold, 1983), but remained within the astrophysical area (Benz, 1988). Monaghan (1994) applied it for the first time to incompressible free-surface flows, treating the fluid as weakly compressible with a stiff equation of state[3]. The density fluctuations depended on the *Ma*, and the speed of sound was chosen so that the maximum fluctuations were on the order of 1%. The examples presented included the evolution of an elliptical droplet, a breaking dam, or a wave maker. Colagrossi (2005) and Marrone (2011) studied in detail its application to breaking waves and free-surface impacts with rigid structures, suitable scenarios to show the SPH capabilities. Kiara (2010) delved deeper into the accuracy, consistency, and stability of the SPH method for free-surface flows, with the aim of reducing the undesired spurious solutions introduced purely by the numerical scheme. Takeda et al. (1994) included viscous terms, solving the Navier-Stokes equations for the flow around a cylinder, 2D Poiseuille flow, and 3D Hagen-Poiseuille flow. Cummins and Rudman (1999) enforced incompressibility in SPH by solving the Poisson equation at each time-step to adjust the

---

[3] Variant of SPH referred to as *weakly compressible smoothed particle hydrodynamics (WCSPH)*

velocity field to make it divergence-free[4]. Regarding multiphase flows, Monaghan et al. (1999) applied SPH to model fluid-fluid interaction with small density differences. Colagrossi and Landrini (2003) achieved larger density ratios, simulating air–water impact flows. Turbulent flows have also been modeled with SPH, either directly by including new terms in the formulation (Adami et al., 2012), using Monte-Carlo probability density functions (Welton, 1998), or as sub-grid models within *large eddy simulations (LES)* (Wagner and Liu, 2000) for example. Additionally, Morris (2000) and Nugent and Posch (2000) included surface-tension forces in the SPH framework.

The efficiency and stability of SPH has been one of the main concerns along its development. One of the earliest improvements to the original formulation was proposed by Hernquist and Katz (1989). They combined SPH with hierarchical trees to improve the cost per step to $O(N \log N)$, where N is the number of particles. Unlike in astrophysics, the fluid domain tends to consist mostly of a compact domain, with few particles detaching from it. It is important to choose the appropriate particle resolution to correctly capture the nature of the flow. Adaptive SPH formulations have been proposed to avoid an excessive resolution far from the regions of interest and therefore optimize the computational resources[5]. The total number of particles is lower, although the formulation is more complex with additional algorithms for either modifying the smoothing length (Shapiro et al., 1996) or merging and splitting particles as they move between regions (Lastiwka et al., 2005; Vacondio et al., 2013; Hu et al., 2017). Along these lines, Shapiro et al. (1993) and Martel et al. (1994) investigated previously anisotropic smoothing algorithms with ellipsoidal kernels, originally proposed by Bicknell and Gingold (1983), and useful to increase the resolution across the flow streamlines. Finally, the SPH method is ideal for parallel computing because only particles within certain distance interact with each other. The use of GPUs and HPCs has allowed to greatly expand the maximum number of particles achievable (Ferrari et al., 2009; Cercos-Pita, 2015; Crespo et al., 2015), with the record at this time over 1 billion particles[6].

SPH usually relies on explicit schemes to integrate forward in time the Navier-Stokes equations. Likewise finite difference methods, these schemes are unconditionally unstable without the existence of some kind of dissipation (Hirsch, 1990). An artificial viscosity term is therefore added to the momentum equation to ensure stability in inviscid flows. Multiple formulations have been proposed for this term: Monaghan and Gingold (1983); Monaghan (1992, 1997); Takeda et al. (1994); Cleary (1996); Morris et al. (1997); Español and Revenga (2003); Colagrossi et al. (2010, 2011), etc. Macià et al. (2012) show how the artificial viscosity can be related to the physical viscosity. Furthermore, Balsara (1995) and Colagrossi (2005) explored an adaptive artificial viscosity useful to model shocks.

One characteristic instability in SPH, named “tensile instability”, consists of an nonphysical clumping of particles. It has been widely studied (Swegle et al., 1995; Morris, 1996), with solutions involving the SPH kernel functions (Macià et al., 2011b), repulsive forces (Monaghan, 2000), and particle shifting algorithms (Lind et al., 2012; Sun et al., 2017) among others.

SPH formulations are also sensitive to the particle disorder (Belytschko et al., 1998). Colagrossi (2005) showed the second-order convergence of the method when particles are arranged on a regular mesh, but may not converge if they are disordered. Quinlan et al. (2006) studied this issue in

---

[4] Variant of SPH referred to as *incompressible smoothed particle hydrodynamics (ISPH)*

[5] Variant of SPH referred to as *adaptive smoothed particle hydrodynamics (ASPH)*

[6] Reported by Barreiro et al. (2015), animation can be seen at `https://youtu.be/B8mP9E75D08`

more detail, exploring the benefits of re-normalizing the kernel based on the particles within the smoothing radius. Furthermore, the use of *moving least squares (MLS)* approximations is clearly beneficial to deal with disordered particles (Lancaster and Salkauskas, 1981). When applied to Galerkin formulations solving the weak form of PDEs, it leads to a set of meshless particle methods grouped under the class name *Partition of Unity* (Belytschko et al., 1996). Methods in this set have been successfully applied to simulate the deformation and fracture of solid materials (Libersky and Petschek, 1991; Benz and Asphaug, 1994, 1995; Melenk and Babuška, 1996; Gray and Monaghan, 2004; Das and Cleary, 2010) and solid-fluid interaction (Antoci et al., 2007). Within the SPH frame, the MLS is used as a correction to the kernel to ensure its consistency independently of the particle distribution (Bonet and Lok, 1999; Dilts, 1999; Souto-Iglesias et al., 2013)[7].

Another common issue in SPH is the high-frequency oscillations found in the pressure, density, and velocity fields. Additional correction terms have been proposed to reduce them, among which we can highlight the XSPH (Monaghan, 1989) and the δ-SPH (Antuono et al., 2010, 2012; Cercos-Pita, 2016; Cercos-Pita et al., 2016). The first one is applied to the velocity field, while the second acts as an additional term in the continuity equation. Colagrossi and Landrini (2003) proposed a periodic re-initialization of the density field using an MLS interpolant to smooth the pressure field.

Gingold and Monaghan (1982) showed how the original SPH formulation could be adapted to conserve linear and angular momentum. Later on, Di Lisio et al. (1997, 1998) and Moussa and Villa (2000) investigated the convergence of the method. The inclusion of the aforementioned numerical correction terms adds an additional challenge to the conservation properties (Morris et al., 1997). Ideally, they should be incorporated to a Lagrangian term from which the equations of motion are derived, which will ensure the conservation of momentum and energy (Bonet and Lok, 1999), but such is not always the case. The energy balance itself has been subject of analysis in the literature (Antuono et al., 2015; Cercos-Pita et al., 2017).

The enforcement of solid-boundary conditions, recognized as one of the main issues of SPH, is the main area of research of this thesis. Monaghan (1994) proposed discretizing the boundaries into particles with repulsive forces. Libersky et al. (1993), Morris et al. (1997) and Colagrossi and Landrini (2003) use “ghost” particles, i.e. layers of particles within the solid domain that mirror the fluid domain and whose properties are adjusted to satisfy the boundary conditions. However, the application of this method with complex geometries and extension to 3D problems is still problematic, with various extensions proposed in the literature (Yildiz et al., 2009; Marrone et al., 2011; De Leffe et al., 2011; Macià et al., 2011a). Other alternatives have been published that explore calculating the intersection of the kernel and the boundary (Feldman and Bonet, 2007), or extend the immersed boundary technique used in finite difference schemes (Hieber and Koumoutsakos, 2008) or the normal flux technique from finite volume schemes (Marongiu et al., 2008; De Leffe et al., 2009).

This summary shows the main contributions that have established the foundation of the SPH method. Multiple additional variants of its formulation have been proposed in the last couple of decades that attempt to yield better stability, performance, and/or precision for different applications. The reader is referred to the publications by Monaghan (2005b, 2012); Liu and Liu (2003); Li and Liu (2004); Violeau (2012) for a more comprehensive review.

---

[7] Variant of SPH referred to as *corrected smoothed particle hydrodynamics (CSPH)*

1.3.3 COUPLING

Recent examples have explored the coupling of SPH with mesh-based finite volume method solvers (Marongiu et al., 2010; Marrone et al., 2016; Kumar et al., 2015; Napoli et al., 2016) and have yielded promising results. An additional appealing approach is to couple SPH with another Lagrangian method with some improved characteristics, like PFEM[8] for example (Idelsohn et al., 2004; Oñate et al., 2008). However, the fully Lagrangian version of Voronoi particle hydrodynamics, VPH (Hess and Springel, 2010), emerges as an even more attractive option for this kind of coupling because its formulation is much closer to SPH. VPH evolved from the Voronoi dynamics method created by Serrano and Español (Serrano and Español, 2001; Serrano, 2006), who developed a meshless scheme that is first-order consistent for the first-order differential operators, regardless of the geometrical distribution of the particles. The method was conceived for molecular dynamics simulations, and entropy evolution equation was part of the formulation. It is worth noticing that SPH itself has also been compared to molecular dynamics (Hoover, 1998).

Voronoi diagrams are named after the Russian and Ukrainian mathematician Georgy Voronoi, who formally defined them in 1908 (Voronoi, 1908) for D-dimensional spaces. They are widely used in fields such engineering, geography, biology, computer graphics, etc. As indicated in Sec. 1.2, they partition the domain into cells based on a set of points or seeds, where each cell encloses the volume closer to one seed than to any other. By brute force, their computation is $O\left(N^2 \log N\right)$, where N is the number of seeds. However, it can be shown how this can be improved at best to $O(N \log N)$. The Fortune's algorithm achieves it in 2D (Fortune, 1987), being therefore optimal and among the most used ones. In 3D and higher dimensions, generating the Voronoi tessellation is more complex (Cignoni et al., 1998). Instead, it is more efficient to determine the Delaunay triangulation and extract the Voronoi tessellation from it, since both structures are dual (Bowyer, 1981; Watson, 1981)[9]. There are several open-source implementations available, among which we can highlight qHull[10] (Barber et al., 1996) and Voro++[11] (Rycroft, 2009). qHull allows computing the Voronoi diagrams in any arbitrary number of dimensions, and is included in Matlab and Python distributions. Moreover, PARAVT[12] is an open source parallell implementation developed by González (2016) that optimizes the use of qHull for large particle sets. On the other hand, Voro++ is restricted to 3D but was developed specifically for science, physics and engineering applications, where the emphasis is placed on the cell properties rather than the complete diagram. Voronoi diagrams are an extensive topic in themselves. They have been generalized to use other metrics apart from the Euclidean distance; to take as seeds objects other than points like line segments; or to weigh differently each seed like power diagrams (Aurenhammer, 1987). Their analysis falls beyond the scope of this work, and the reader is referred to the detailed studies done by Okabe et al. (2000), Okabe (2016), and De Berg et al. (2008).

Within the SPH framework, the application of Voronoi diagrams has been quite recent. Shobeyri and Ardakani (2017) proposed using Voronoi diagrams to improve the computation of the second order derivatives, Ghaffari and Xiao (2016) used centroid Voronoi tessellations to adjust the particle

---

[8] `http://www.cimne.com/pfem/`

[9] Bowyer-Watson algorithm

[10] `http://www.qhull.org`

[11] `http://math.lbl.gov/voro++/`

[12] `https://github.com/regonzar/paravt`

positions and improve the consistency of the method, and Chiaki and Yoshida (2015) explored using them as part of a particle splitting algorithm. The idea of a hybrid Lagrangian Voronoi–SPH scheme was first proposed by Barcarolo (2013) and Barcarolo et al. (2014). They used a finite volume formulation for the Voronoi sub-domain and Riemann-SPH for the SPH sub-domain. The coupling was achieved by considering the SPH particles as Voronoi ones when interacting with Voronoi particles, and vice versa. This coupling does not preserve the order of either method when taken to the particle level, and some improvements are required.

The SPH European Research Interest Community (SPHERIC), which has grown to bring together most of the researchers working on SPH, identifies the following remaining SPH Grand Challenges[13]:

1. Convergence, consistency, and stability.
2. Boundary conditions.
3. Adaptivity.
4. Coupling to other methods.
5. Applicability to industry.

The use of Voronoi diagrams to improve the enforcement of wall boundary conditions is directly linked to the 1st, 2nd, and 4th challenges.

1.4 MAJOR CONTRIBUTIONS

The main contribution of this work is the foundation of a fully Lagrangian VPH-SPH coupled scheme. The diffusive term in the continuity equation from the δ-SPH formulation has been extended to the VPH, hence the δ-VSPH name. As explained in detail in Ch. 3, the fluid domain is subdivided into Voronoi and SPH sub-domains based on the distance to the boundaries, with certain overlapping between both. All particles in the Voronoi sub-domain are used to generate a Voronoi diagram at each time step, so VPH can be used to compute the dynamics. We can therefore distinguish three separate regions:

- Voronoi: Zone close to the solid boundaries, where only VPH is used in order to accurately implement the solid boundary conditions.
- Buffer: Overlapping zone between both sub-domains that allows for a smooth transition between methods. All related variables are computed in this region both with the SPH and VPH formulations and combined using a weighting function.
- SPH: Rest of the fluid domain that is not influenced by the solid boundaries, where only SPH is used for its simplicity and efficiency to deal with free-surfaces.

The main advantage of this approach is that it is targeted to the core formulation of SPH. VPH and SPH are so similar that most of the improvements proposed in the literature for SPH can be directly extended to VPH. In the end, both methods provide a way to approximate the spatial

---

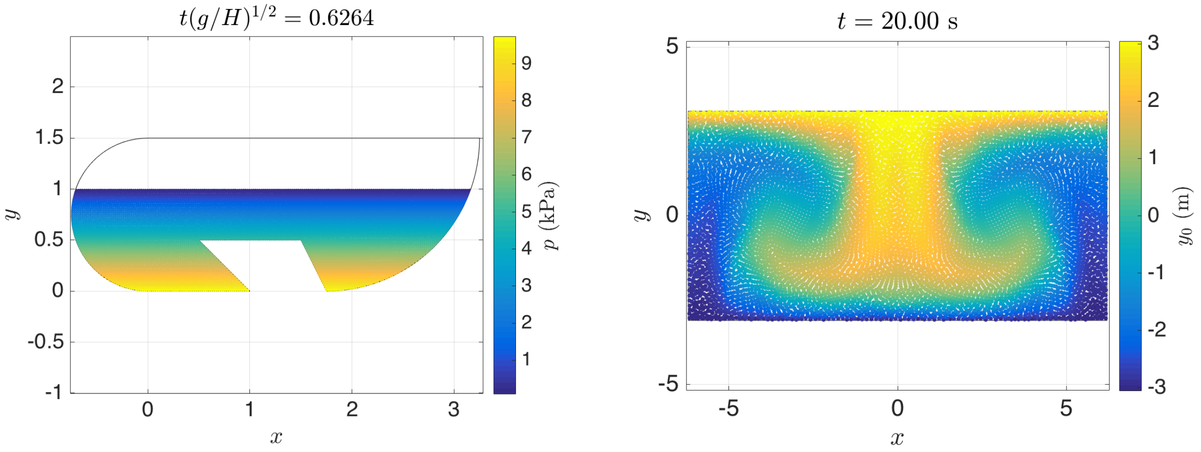
[13] http://spheric-sph.org/grand-challenges

derivatives without a predefined connectivity between particles. Furthermore, the Voronoi tessellation provides additional information beneficial to the SPH scheme itself. Firstly, it allows us to associate a certain volume to each particle, and considering its mass, a derived density. These values can be used for the density re-initialization to reduce the high-frequency pressure oscillations. Also, they allow enforcing the conservation of total volume in addition to the conservation of mass in the Voronoi region. In addition, the Voronoi diagram can allow us to detect unphysical clustering and avoid the tensile instability.

Unlike ghost particles, Voronoi diagrams can be simply extended to 3D and even higher dimensions. They can also be applied to complex geometries, although the existence of concave faces requires an additional trimming process that can be computationally expensive. To avoid excessive complexity, the size of the boundary faces is used to set an upper bound to the maximum size of the surrounding particles.

The main source of difficulties for this type of coupling comes from the different characteristic lengths of each method. SPH relies on all the particles within a given smoothing radius, typically 4 times the particle size, while VPH uses only the immediate neighbors in contact with the Voronoi cell. This fact, which is precisely what simplifies the enforcement of the boundary conditions, requires also a larger buffer zone to keep the scheme conservative and to avoid non-physical interactions between the methods.

In addition to the δ-SPH correction mentioned above, the density field is reinitialized following Colagrossi and Landrini (2003), and a particle shifting algorithm inspired by Lloyd (1982) is included in the VPH formulation to avoid an excessive cell distortion. The resulting scheme has been tested under configurations where analytic solutions are known (Fig. 1.5), showing good agreement. Being an explicit scheme, the code was developed keeping in mind the inclusion of parallel computing. However, the ability to use multiple computing nodes hasn't been implemented yet. The emphasis was placed in proving the feasibility of the method, which limited the number of particles used in the simulations.



(a) Hydrostatic equilibrium with concave boundaries. Particles colored based on their gauge pressure

(b) Two Lamb-Osseen contra-rotating vortices. Particles colored based on their initial vertical position

Figure 1.5: Simulation results for cases with known analytic solutions

1.5 THESIS STRUCTURE

This thesis is structured as follows:

1. The continuous model is presented, discussing the governing differential equations and their underlying assumptions *(chapter 2)*. The specific characteristics of the weakly compressible fluid model are included.

2. The detailed explanation of the coupled scheme constitutes the main part of this thesis *(chapter 3)*. A description of the main characteristics of SPH and VPH is included, followed by their coupling strategy. The way boundary conditions are implemented deserves a dedicated analysis that follows next *(section 3.4)*. Finally, the specifics of the density re-initialization, and particle shifting algorithms are presented *(sections 3.5-3.6)*.

3. The next step is presenting the actual implementation of the coupled scheme using C++ and Matlab *(chapter 4)*. Special emphasis is placed on describing the leap-frog algorithm used to march forward in time, the neighboring boundaries and particles search algorithm, and the initialization process.

4. The energy balance of the scheme is discussed thereafter *(chapter 5)*.

5. The method is then tested with some verification cases: hydrostatic equilibrium with simple and complex geometries; reflecting sound wave; steady and starting Couette flow; decaying Lamb-Ossen vortex; two contra-rotative Lamb-Ossen vortices; and an impinging jet *(chapter 6)*.

6. Conclusions and future lines of work are enumerated to close the thesis *(chapter 7)*.

CHAPTER 2

CONTINUUM MODEL

The scope of this research covers monophasic free-surface viscous flows. This chapters summarizes their governing differential equations (Sec. 2.1), which will be solved numerically later. Special emphasis is placed in describing the weakly compressible equation of state used (Sec. 2.2), and the selection of an adequate speed of the sound to approximate properly incompressible fluids while keeping an acceptable computational cost (Sec. 2.3).

2.1 GOVERNING EQUATIONS

Monophasic free-surface viscous flows are governed by the well known Navier–Stokes equations. These equations, derived from the balance of mass, momentum, and energy, read in their general Lagrangian differential form:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho \nabla \cdot \boldsymbol{u} = 0 , \tag{2.1}$$

$$\rho \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \rho \boldsymbol{f} + \nabla \cdot \underline{\underline{\sigma}} , \tag{2.2}$$

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\left(e + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u} \right) = \rho \boldsymbol{f}\cdot\boldsymbol{u} + \nabla \cdot \left(\underline{\underline{\sigma}}\boldsymbol{u} \right) - \nabla \cdot \boldsymbol{q} + \rho \dot{Q} , \tag{2.3}$$

where $\mathrm{d}/\mathrm{d}t$ is the Lagrangian time derivative, ρ the fluid density, t the time, $\boldsymbol{u}$ the fluid velocity vector, $\boldsymbol{f}$ the body forces per unit mass, $\underline{\underline{\sigma}}$ the stress tensor, e the internal energy per unit mass, $\boldsymbol{q}$ the heat flux vector, and $\dot{Q}$ the volumetric heat deposition.

The long-term applications that motivate the present research, such as additive manufacturing, mostly involve liquids. Moreover, the expected fluid regimes allow us to accurately model them as barotropic and weakly compressible. Under this assumption, and by neglecting entropy influence, the pressure becomes a function of density only in an equation of state, and the energy equation (2.3) becomes uncoupled from the mass and momentum balances in Eqs. (2.1)-(2.2).

In this first stage, thermal effects have not been included so the energy equation can be dropped to simplify the formulation. Also, only uniform gravitational body forces and Newtonian fluids are considered. Under these conditions,

$$\underline{\underline{\sigma}} = (-p + \lambda \nabla \cdot \boldsymbol{u})\, \mathbb{I} + 2\mu \mathbb{D} , \tag{2.4}$$

$$\rho \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \rho \boldsymbol{g} - \nabla p + \nabla (\lambda (\nabla \cdot \boldsymbol{u})) + \nabla \cdot (2\mu \mathbb{D}) , \tag{2.5}$$

where $\boldsymbol{g}$ is the gravity acceleration vector, p the pressure, μ and λ the first and second coefficients of viscosity, and $\mathbb{D}$ the rate of deformation tensor defined as

$$\mathbb{D} = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{T} \right) , \tag{2.6}$$

with the superscript ^{T} indicating tensor transpose. Substituting the definition of $\mathbb{D}$ in Eq. (2.5), using the algebraic relation $\nabla\cdot(\nabla\boldsymbol{u})^T = \nabla(\nabla\cdot\boldsymbol{u})$, and with the additional hypothesis of constant dynamic viscosity coefficients λ and μ , we finally arrive at

$$\rho\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \rho\boldsymbol{g} - \nabla p + (\lambda+\mu)\nabla(\nabla\cdot\boldsymbol{u}) + \mu\nabla^2\boldsymbol{u}\,, \tag{2.7}$$

where $\nabla^2\boldsymbol{u} = \nabla\cdot(\nabla\boldsymbol{u})$ is the Laplacian differential operator. The compressible viscosity term is negligible in the weakly compressible regime for the flows studied in this work (see, *e.g.*, Marrone et al. (2013) and Colagrossi et al. (2013)) and it is not further considered.

Regarding the equation of state, the formulation proposed by Monaghan (1994) is followed. Section 2.2 will discuss it in detail. The relaxation from incompressible to weakly compressible is generally followed within the SPH method. This way, we can avoid solving the Poisson equation for the pressure field, and use an explicit time integration for the discrete equations.

Based on all these hypotheses, we can simplify the previous expressions to obtain the final governing and constitutive equations of the continuum model:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho\,\nabla\cdot\boldsymbol{u}\,, \tag{2.8}$$

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{g} - \frac{\nabla p}{\rho} + \nu\,\nabla^2\boldsymbol{u}\,, \tag{2.9}$$

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{u}\,, \tag{2.10}$$

$$p = \frac{c^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] + p_0\,. \tag{2.11}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. The fluid velocity, $\boldsymbol{u}$, is defined as the material derivative of a fluid material point at the position $\boldsymbol{r}$. c is the speed of the sound discussed in Sec. 2.3, and γ is the coefficient that determines the level of compressibility of the fluid, discussed in detailed in Sec. 2.2. p_0 and ρ_0 are the reference pressure and density of the fluid, respectively.

Finally, the following possible boundary conditions (BC's) are considered:

Solid boundary (wall): Two possible alternatives can be selected

- No slip BC: Fluid velocity on the boundary is equal to the velocity of the boundary. Typical in viscous shear dominant problems.

$$\boldsymbol{u}|_B = \boldsymbol{u}_B\,. \tag{2.12}$$

- Slip BC: Tangential velocity of the fluid is independent of the boundary, typical for inviscid or very thin boundary layers problems. Only normal velocity is equal to the wall velocity, *i.e.* the fluid cannot penetrate the boundary.

$$\left.\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{n}}\right|_B = 0 \quad\Longrightarrow\quad \boldsymbol{u}\cdot\boldsymbol{n}_B = \boldsymbol{u}_B\cdot\boldsymbol{n}_B\,. \tag{2.13}$$

where $\boldsymbol{n}_B$ represents the normal vector to the boundary pointing towards the fluid. Note that this condition can also be used to model symmetry planes (Slip BC).

Free surface: Pressure at the free surface is the atmospheric reference pressure. No surface tension has been included yet.

$$p_{FS} = p_0 \,. \tag{2.14}$$

Note that Lagrangian particles follow the free surface shape even for large deformations fulfilling implicitly the kinematic BC.

Periodic:

$$\boldsymbol{u}(\boldsymbol{r})|_B = \boldsymbol{u}(\boldsymbol{r} + \Delta\boldsymbol{r}) \,, \quad p(\boldsymbol{r})|_B = p(\boldsymbol{r} + \Delta\boldsymbol{r}) \,. \tag{2.15}$$

Inlet/Outlet (I/O): Boundaries through which the fluid enters or leaves the domain. Dirichlet or Neumann BC's can be assigned to the velocity and pressure variables depending on the boundary nature. If no specific condition is imposed for a variable, the natural Neumann BC is assumed ($\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{n}}\big|_{IO} = 0$). The following combinations are considered:

- Velocity I/O: Fluid velocity is fixed at the boundary as $\boldsymbol{u}_{IO}$,

$$\boldsymbol{u}|_{IO} = \boldsymbol{u}_{IO} \,, \quad \left.\frac{\mathrm{d}p}{\mathrm{d}\boldsymbol{n}}\right|_{IO} = 0 \,. \tag{2.16}$$

- Pressure I/O: Pressure is fixed at the boundary as p_{IO},

$$\left.\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{n}}\right|_{IO} = 0 \,, \quad p|_{IO} = p_{IO} \,. \tag{2.17}$$

- Free I/O: No specific conditions imposed,

$$\left.\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{n}}\right|_{IO} = 0 \,, \quad \left.\frac{\mathrm{d}p}{\mathrm{d}\boldsymbol{n}}\right|_{IO} = 0 \,. \tag{2.18}$$

2.2 WEAKLY COMPRESSIBLE EQUATION OF STATE

The purpose of this section is to clarify the equation of state used, anticipated in Eq. (2.11). We start by acknowledging that the use of particles introduces numerical variations in the density which should be taken into account if a compressible model is used. Incompressibility can be imposed by determining a pressure correction term that should satisfy a Poisson equation in each time step, although this process increases the computational cost of the simulations.

A more efficient approach consists of letting the density fluctuate and using an equation of state to determine the consequent pressure variations, as proposed by Monaghan (1994). For liquids we can use a weakly compressible equation of state following Tait's equation, which enforces very low density variations and is efficient to compute:

$$p = B_p \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] + p_0 \,, \tag{2.19}$$

where B_p and γ are the constants that control the amplitude of the pressure change due to the density fluctuations. We can characterize the relative density fluctuations as a function of a parameter η, defined as

$$\eta = \frac{\Delta\rho}{\rho} \approx \frac{u^2}{c^2} \,, \tag{2.20}$$

where $u = \|\boldsymbol{u}\|_2$ is the velocity magnitude given by its Euclidean norm. Typically $\eta = 0.01$, allowing density fluctuations of the order of 1%.

The corresponding pressure fluctuations can be determined from Eq. (2.19).

$$\begin{aligned}
\Delta\rho &= \rho_2 - \rho_1\,, && (2.21)\\
\Delta p &= p_2 - p_1 = B_p\left[\left(\frac{\rho_2}{\rho_0}\right)^{\gamma} - \left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\right] = B_p\left[\left(\frac{\rho_1+\Delta\rho}{\rho_0}\right)^{\gamma} - \left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\right]\\
&= B_p\left[\left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\left(1+\frac{\Delta\rho}{\rho_1}\right)^{\gamma} - \left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\right]\\
&= B_p\left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\left[(1+\eta)^{\gamma}-1\right]. && (2.22)
\end{aligned}$$

Using a Taylor expansion of $(1+\eta)^{\gamma}$ around the equilibrium ($\eta = 0$) leads to

$$(1+\eta)^{\gamma} \approx \left[(1+\eta)^{\gamma}\right]_{\eta=0} + (\eta-0)\left[\gamma(1+\eta)^{\gamma-1}\right]_{\eta=0} = 1+\gamma\eta\,, \tag{2.23}$$

$$\Delta p \approx B_p\left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\left[1+\gamma\eta-1\right] = B_p\left(\frac{\rho_1}{\rho_0}\right)^{\gamma}\gamma\eta\,. \tag{2.24}$$

Both ρ_1 and ρ_2 oscillate closely around ρ_0, which entitle us to approximate $\frac{\rho_1}{\rho_0} \approx 1$ to obtain the order of magnitude of the density and pressure variations:

$$\Delta\rho \approx \rho_0\eta\,, \tag{2.25}$$

$$\Delta p \approx B_p\gamma\eta\,. \tag{2.26}$$

Using the definition of the speed of sound we can relate the two magnitudes above, and express B_p as a function of the other variables:

$$c^2 = \frac{\partial p}{\partial\rho} \approx \frac{\Delta p}{\Delta\rho} = \frac{B_p\gamma\eta}{\rho_0\eta} = \frac{B_p\gamma}{\rho_0}\,, \tag{2.27}$$

$$B_p = \frac{c^2\rho_0}{\gamma}\,. \tag{2.28}$$

Substituting Eq. (2.28) in Eq. (2.19) we obtain the weakly compressible equation of state (2.11), dependent only on γ:

$$p = \frac{c^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] + p_0\,.$$

As discussed by Becker and Teschner (2007), typically $\gamma = 7$. However, if we analyze the impact that the variation of each variable in this equation of state has, we can observe how c is more relevant than γ. As a matter of fact, the order of magnitude of the pressure fluctuations determined in Eq. (2.26) can be rewritten expanding B_p from Eq. (2.28) as

$$\Delta p \approx \frac{c^2\rho_0}{\gamma}\gamma\eta = c^2\rho_0\eta\,, \tag{2.29}$$

showing how it does not depend on γ. We can therefore choose $\gamma = 1$ without a major impact on the pressure fluctuations and obtain a simpler equation of state, as proposed by Antuono et al. (2010):

$$p = c^2 (\rho - \rho_0) + p_0 . \tag{2.30}$$

Furthermore, it is not necessary to use the physical speed of sound, since the compressibility effects should be physically negligible for the studied problems under the weakly compressible assumption (Mach number, $Ma = u/c << 1$). Consequently, we can vary c to adjust the level of compressibility based on the flow conditions, as analyzed below in Sec. 2.3. The common practice in SPH is to reduce c as much as possible to avoid the extremely small time steps that would result from using the physical sound velocity, as will be shown later in Sec. 4.5.

2.3 SPEED OF SOUND, c

As stated before, the selection of c is critical to determine the time step for the stability of the scheme. Physically, we can interpret it as the speed at which information propagates through the fluid. However, when the time-scale of the analyzed problem is much larger than this transfer of information ($Ma << 1$), we can modify the value of c, up to a certain point, with negligible consequences in the flow response. It is a common practice to set the upper limit at $Ma = 0.1$, which keeps density fluctuations below 1% ($\eta = \Delta\rho/\rho_0 < 0.01$, see Eq. (2.20)), and where the weakly compressible assumption remains valid.

In general, the common practice in SPH is to use a fictitious c one or two orders of magnitude smaller in order to achieve the largest possible time step while keeping $\eta < 0.01$. Because the velocity field is unknown a priori, it should be checked during the simulation that this condition is met. However, the following considerations are suggested to choose the value of c beforehand, based on an order of magnitude analysis of the characteristic variables of the flow:

Velocity driven flows such as uniform stream or flow around objects

$$\begin{aligned} &u_{max} \approx u_0 , \\ &\Delta p \approx c^2 \Delta\rho \approx \rho u^2 \quad \rightarrow \quad c^2 = \frac{\rho u^2}{\Delta\rho} > \frac{u_0^2}{0.01} \rightarrow c > 10\, u_0 . \end{aligned} \tag{2.31}$$

Gravity driven flows such as gravity waves or hydrostatic conditions

$$\begin{aligned} &\Delta p \approx \rho g H , \\ &\Delta p = c^2 \Delta\rho = \rho g H \quad \rightarrow \quad c^2 = \frac{\rho g H}{\Delta\rho} > \frac{gH}{0.01} \rightarrow c > 10\sqrt{gH} , \end{aligned} \tag{2.32}$$

where H is the reference height.

Pressure driven flows such as Poiseuille flows

$$\begin{aligned} &\|\nabla p\| \approx K , \\ &\|\nabla p\| \approx \frac{\Delta p}{\Delta x} = \frac{c^2 \Delta\rho}{\Delta x} = K \quad \rightarrow \quad c^2 = K\frac{\Delta x}{\Delta\rho} > K\frac{\Delta x}{0.01\rho} , \end{aligned} \tag{2.33}$$

where Δx is the particle size, which gives the order of magnitude of the particle spacing.

Shear driven flows such as Couette flows

$$\mu\nabla^2\boldsymbol{u} \approx \nabla p \quad \rightarrow \quad \mu\frac{\Delta u}{\Delta x^2} \approx \frac{\Delta p}{\Delta x}\,, \tag{2.34}$$

$$\Delta p = c^2\Delta\rho = \mu\frac{\Delta u}{\Delta x} \quad \rightarrow \quad c^2 = \frac{\mu}{\Delta\rho}\frac{\Delta u}{\Delta x} > \frac{\mu}{0.01\rho}\frac{\Delta u}{\Delta x}\,. \tag{2.35}$$

Note that in this last case, the condition requires an estimate of the maximum velocity gradient on the flow, similar to the pressure gradient previously. In both cases, if transitory conditions are studied, the maximum gradients during the transition phase should be used, which are generally significantly larger than in the steady state.

CHAPTER 3

METHOD

This chapter contains a detailed description of the coupled scheme. First, it presents each numerical method individually: SPH in Sec. 3.1, VPH in Sec. 3.2, and the coupling strategy in Sec. 3.3. The enforcement of different types of boundary conditions are discussed next in Sec. 3.4. Special emphasis is placed in the solid boundary conditions, both with convex and concave walls (Secs. 3.4.2 and 3.4.3, respectively). A new free-surface treatment under the VPH method is proposed in Secs. 3.4.4-3.4.5, with periodic boundaries and inlet/outlets discussed afterwards in Secs. 3.4.6-3.4.7. Finally, two additional numerical corrections to improve the stability of the method are analyzed: density re-initialization (Sec. 3.5) and particle shifting (Sec. 3.6).

3.1 SPH

3.1.1 FUNDAMENTALS

The SPH method can be explained as a two-step approximation method, as originally described by Libersky et al. (1993), that involves:

1. Smoothing: Any field variable at a given point can be approximated as the weighted average of its value around this point.

2. Particle approximation: The continuous domain is discretized into a set of points, named particles, on which mass is lumped.

We can always re-write any field variable $f(\boldsymbol{r})$ exactly as

$$f(\boldsymbol{r}) = \int_{\Omega} f(\boldsymbol{r}^*)\, \delta(\boldsymbol{r} - \boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* \,, \tag{3.1}$$

where Ω is the fluid domain, $\boldsymbol{r}^*$ is a local integration variable, $d\mathcal{V}^*$ a differential volume, and $\delta(\boldsymbol{r} - \boldsymbol{r}^*)$ is the Dirac delta function. Its smoothed approximated value is obtained by using a known finite weighting function W, *i.e.* kernel, instead of the Dirac delta.

$$< f(\boldsymbol{r}) > = \int_{\Omega} f(\boldsymbol{r}^*)\, W(\boldsymbol{r} - \boldsymbol{r}^*, h)\, \mathrm{d}\mathcal{V}^* \,, \tag{3.2}$$

where h is the smoothing length that characterizes the radius of action of the kernel. The benefit of the smoothing comes when evaluating the spatial derivatives with integration by parts:

$$\begin{aligned} < \nabla_{r} f(\boldsymbol{r}) > &= \int_{\Omega} [\nabla_{\boldsymbol{r}^*} f(\boldsymbol{r}^*)]\; W(\boldsymbol{r} - \boldsymbol{r}^*, h)\, \mathrm{d}\mathcal{V}^* \\ &= \int_{\partial\Omega} f(\boldsymbol{r}^*) W(\boldsymbol{r} - \boldsymbol{r}^*, h)\, \mathbf{d}\boldsymbol{S}^* \; - \int_{\Omega} f(\boldsymbol{r}^*)\, \nabla_{\boldsymbol{r}^*} W(\boldsymbol{r} - \boldsymbol{r}^*, h)\, \mathrm{d}\mathcal{V}^* \,, \end{aligned} \tag{3.3}$$

where $\partial\Omega$ is the perimeter of the domain of integration and $\mathbf{d}\boldsymbol{S}^*$ a differential surface normal vector. If we impose the condition that the kernel is zero in the far field, the first term vanishes. Furthermore, if we also require the kernel function to be symmetric, $W(\boldsymbol{r}-\boldsymbol{r}^*) = W(\boldsymbol{r}^*-\boldsymbol{r}) \rightarrow \nabla_{\boldsymbol{r}^*} W(\boldsymbol{r}-\boldsymbol{r}^*) = -\nabla_{\boldsymbol{r}^*} W(\boldsymbol{r}^*-\boldsymbol{r}) = \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*)$ as long as h is constant or at least symmetric between $\boldsymbol{r}$ and $\boldsymbol{r}^*$, *i.e.* $h(\boldsymbol{r}) = h(\boldsymbol{r}^*)$. This fact allows us to arrive at the final expression for the derivative

$$< \nabla_{\boldsymbol{r}} f(\boldsymbol{r}) > \; = \int_\Omega f(\boldsymbol{r}^*)\, \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*, h)\, \mathrm{d}\mathcal{V}^* \,. \tag{3.4}$$

We have successfully expressed the derivative as a function of the variable itself and the derivative of the kernel which is a known function.

We can now proceed with the second step, the particle approximation. By going from the continuous to the discrete level, we can simplify the previous integrals into summations since the values of f and W are evaluated only at the particle locations.

$$< f(\boldsymbol{r}) > \; = \sum_b \int_{\mathcal{V}_b} f(\boldsymbol{r}_b)\, W(\boldsymbol{r}-\boldsymbol{r}_b, h)\, \mathrm{d}\mathcal{V}^* \; = \sum_b f(\boldsymbol{r}_b)\, W(\boldsymbol{r}-\boldsymbol{r}_b, h)\, \mathcal{V}_b \,, \tag{3.5}$$

$$< \nabla_{\boldsymbol{r}} f(\boldsymbol{r}) > \; = \sum_b \int_{\mathcal{V}_b} f(\boldsymbol{r}_b)\, \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}_b, h)\, \mathrm{d}\mathcal{V}^* \; = \sum_b f(\boldsymbol{r}_b)\, \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}_b, h)\, \mathcal{V}_b \,, \tag{3.6}$$

where the counter b runs through the particles that fall within the smoothing domain with non-zero W. The particles themselves are just interpolation nodes, but we can define their associated volume based on the mass lumped on them as $\mathcal{V}_b = \frac{m_b}{\rho_b}$, with m being the mass and ρ the density.

Note that even if f is discrete, $< f >$ is still a continuous value. However, for computation purposes we are usually interested only on evaluating the smoothed values on the location of the particles themselves. To simplify the notation, the common convention in SPH is adopted:

- a for the analyzed particle, and b for all particles within the smoothing domain
- $< f(\boldsymbol{r}_a) >= f_a$
- $f_{ab} = f_a - f_b$
- $\overline{f}_{ab} = (f_a + f_b)/2$
- $W(\boldsymbol{r}_a - \boldsymbol{r}_b, h) = W_{ab}$
- $\nabla_{\boldsymbol{r}_a} W(\boldsymbol{r}_a - \boldsymbol{r}_b, h) = \nabla_a W_{ab}$

We can therefore approximate the values at the particle locations based on the surrounding ones as

$$f_a = \sum_b \frac{m_b}{\rho_b} f_b\, W_{ab} \,, \tag{3.7}$$

$$\nabla f_a = \sum_b \frac{m_b}{\rho_b} f_b\, \nabla_a W_{ab} \,. \tag{3.8}$$

However, Eq. (3.8) does not necessarily vanish for constant fields due to the random location of the neighbor particles, which is clearly incorrect. To fix it, we can use an auxiliary function Φ to rewrite the derivative as

$$\begin{aligned}\nabla f_a &= \frac{1}{\Phi_a}\left[\nabla(\Phi f)_a - f_a \nabla\Phi_a\right] = \frac{1}{\Phi_a}\left[\sum_b \frac{m_b}{\rho_b}\Phi_b f_b \nabla_a W_{ab} - f_a \sum_b \frac{m_b}{\rho_b}\Phi_b \nabla_a W_{ab}\right] \\ &= -\frac{1}{\Phi_a}\sum_b \frac{m_b}{\rho_b}\Phi_b (f_a - f_b)\nabla_a W_{ab}\,. \end{aligned} \tag{3.9}$$

Different choices of Φ have been explored in the literature, with the most common ones being

$$\Phi = 1 \quad \rightarrow \quad \nabla f_a = -\sum_b \frac{m_b}{\rho_b} f_{ab}\nabla_a W_{ab}\,, \tag{3.10}$$

$$\Phi = \rho \quad \rightarrow \quad \nabla f_a = -\frac{1}{\rho_a}\sum_b m_b f_{ab}\nabla_a W_{ab}\,, \tag{3.11}$$

These expressions can be directly extended to compute the divergence of a vector field:

$$\Phi = 1 \quad \rightarrow \quad \nabla\cdot \boldsymbol{f}_a = -\sum_b \frac{m_b}{\rho_b} \boldsymbol{f}_{ab}\cdot\nabla_a W_{ab}\,, \tag{3.12}$$

$$\Phi = \rho \quad \rightarrow \quad \nabla\cdot \boldsymbol{f}_a = -\frac{1}{\rho_a}\sum_b m_b \boldsymbol{f}_{ab}\cdot\nabla_a W_{ab}\,, \tag{3.13}$$

The Laplacian operator that shows up in the viscous term involves second derivatives. The same approach can be followed to approximate it, but the resulting formulation has a certain number of disadvantages as it is shown in the literature (see *e.g.* Monaghan (2005b)). An alternative approach is focusing on the formulations proposed for inviscid flows. Even if the Laplacian is not present, an artificial viscosity (AV) term is required in the momentum equation for stability to dissipate the numerical noise. Multiple formulations can be found in the literature, as introduced in Sec. 1.3.2. In this research, we use the Monaghan-Cleary-Gingold's formulation discussed by Macià et al. (2012) that evolved from the original one proposed by Monaghan and Gingold (1983). It conserves linear and angular momentum, vanishes in the limit of $h \rightarrow 0$ and for rigid rotations, and is Galilean invariant (Monaghan, 2005b).

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV,SPH} = -\sum_b m_b \Pi_{ab}\nabla_a W_{ab} \tag{3.14}$$

$$\Pi_{ab} = \frac{\alpha\,\overline{h}_{ab}\,\overline{c}_{ab}\,\overline{\rho}_{ab}}{\rho_a\,\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{||\boldsymbol{r}_{ab}||^2} \tag{3.15}$$

where α is a coefficient used to adjust the magnitude of this artificial viscous term. Note that $\Pi_{ab} = \Pi_{ba}$ and how Eq. (3.15) accounts for possible different smoothing lengths and sound speeds between particles. Appendix A demonstrates how we can trace this term back to the original Navier-Stokes momentum Eq. (2.9), following the work by Español and Revenga (2003) and Violeau (2009). By matching terms, we can link Eq. (3.14) to the actual viscous components and determine an approximation to the Laplacian operator, independent of the kernel used and consistent with Monaghan (2005b), Hu and Adams (2006), and Macià et al. (2011a).

$$\frac{1}{2(D+2)}\,\alpha\, h\, c \;\rightarrow\; \nu\,, \tag{3.16}$$

$$\nabla^2 \boldsymbol{u}_a \approx -\sum_b 2(D+2)\, m_b \frac{\overline{\rho}_{ab}}{\rho_a\,\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\nabla_a W_{ab}\,, \tag{3.17}$$

where D is the number of spatial dimensions.

3.1.2 KERNEL

From the expressions above, it is clear that the kernel function is at the core of the SPH method, and its selection is crucial to obtain valid results. In general, no direction is preferred, so the kernel can be expressed as a function of a normalized radial coordinate, q.

$$q = \frac{\|\boldsymbol{r}-\boldsymbol{r}^*\|}{h}\,, \tag{3.18}$$

$$W(\boldsymbol{r}-\boldsymbol{r}^*, h) \;\rightarrow\; W(q) \;=\; \frac{k_k}{h^D} M(q)\,, \tag{3.19}$$

where k_k is a constant coefficient so the kernel is normalized (Eq. (3.24)). Section A.3 shows how k_k in its general form can be written as

$$k_k = \frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{D/2}\int_\Omega M(q)\, q^{D-1}\mathrm{d}q}\,, \tag{3.20}$$

where Γ is the Gamma function. Regarding the kernel gradient, we can expand Eq. (3.18) such that

$$\nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*, h) \;=\; \frac{k_k}{h^D}\frac{\mathrm{d}M(q)}{\mathrm{d}q}\nabla_{\boldsymbol{r}} q = \frac{k_k}{h^D}\frac{\mathrm{d}M(q)}{\mathrm{d}q}\frac{(\boldsymbol{r}-\boldsymbol{r}^*)}{h^2 q}\,, \tag{3.21}$$

which is usually expressed as

$$\nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*, h) = (\boldsymbol{r}-\boldsymbol{r}^*) F(q)\,, \tag{3.22}$$

$$F(q) = \frac{k_k}{h^{D+2}\, q}\frac{\mathrm{d}M(q)}{\mathrm{d}q}\,. \tag{3.23}$$

The form of the kernel function is not arbitrary, but should satisfy a minimum set of requirements (Liu and Liu, 2003):

1. It should be normalized:

$$\int_\Omega W(\boldsymbol{r}-\boldsymbol{r}^*, h)\mathrm{d}\mathcal{V}^* = 1\,. \tag{3.24}$$

2. It should be compactly supported, *i.e.*

$$W(\boldsymbol{r}-\boldsymbol{r}^*, h) = 0 \quad \forall\, \|\boldsymbol{r}-\boldsymbol{r}^*\| > \kappa h\,, \tag{3.25}$$

where κ is used to adjust the spread of the smoothing function. Therefore, $q \in [0, \kappa]$ for $W \neq 0$.

3. It should be positive within the support domain

$$W(\boldsymbol{r}-\boldsymbol{r}^{*},h) > 0 \quad \forall q \leq \kappa \,. \tag{3.26}$$

4. It should decrease monotonically, so the impact weight of particles decrease as they lay further from the analyzed location

$$F(q) < 0 \quad \forall q \leq \kappa \,. \tag{3.27}$$

5. It should converge to the Dirac delta function in the limit of $h \to 0$

$$\lim_{h\to 0} W(\boldsymbol{r}-\boldsymbol{r}^{*},h) = \delta(\boldsymbol{r}-\boldsymbol{r}^{*}) \,. \tag{3.28}$$

6. It should be an even function, *i.e.* a symmetric function, so

$$W(\boldsymbol{r}-\boldsymbol{r}^{*},h) = W(-\boldsymbol{r}+\boldsymbol{r}^{*},h) \,, \tag{3.29}$$

   which is automatically satisfied when expressed as a function of $M(q)$.

7. It should be sufficiently smooth *i.e.* have continuous derivatives. The higher order of continuity, the better to approximate higher derivatives of a function and be less sensitive to particle disorder.

Multiple kernel formulations can be found in the literature, and we refer the reader to Liu and Liu (2003) and Violeau (2012) for a detailed analysis of the most commonly used. In this research, the C2 Wendland kernel (WC2) is used following Macià et al. (2011b), which has a $2h$ compact support and the following formulation (Fig. 3.1):

$$M(q) = \begin{cases} (1.0-0.5\,q)^4\,(1.0+2.0\,q) & q \leq 2 \\ 0 & q > 2 \end{cases} \,, \tag{3.30}$$

$$F(q) = \begin{cases} -\dfrac{k_k}{h^{D+2}}\,5.0\,(1.0-0.5q)^3 & q \leq 2 \\ 0 & q > 2 \end{cases} \,, \tag{3.31}$$

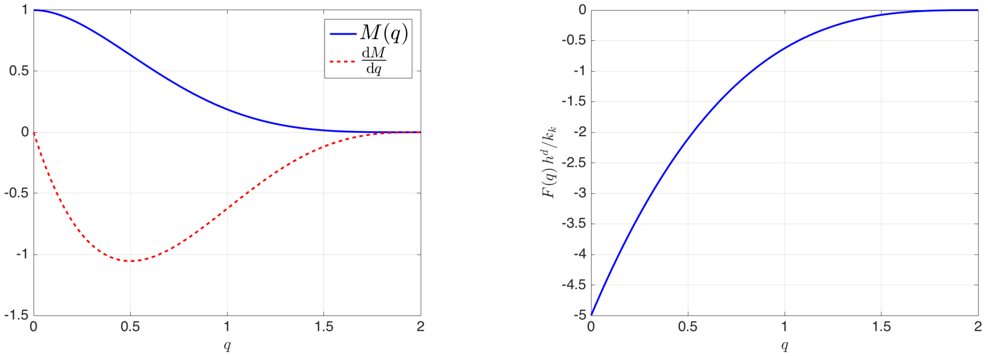
that leads to its dimensional form (Fig. 3.2)

$$W(\boldsymbol{r}-\boldsymbol{r}^{*},h) = \begin{cases} \dfrac{k_k}{h^D}\left(1.0-0.5\,\dfrac{\|\boldsymbol{r}-\boldsymbol{r}^{*}\|}{h}\right)^4\left(1.0+2.0\,\dfrac{\|\boldsymbol{r}-\boldsymbol{r}^{*}\|}{h}\right) & \|\boldsymbol{r}-\boldsymbol{r}^{*}\| \leq 2h \\ 0 & \|\boldsymbol{r}-\boldsymbol{r}^{*}\| > 2h \end{cases} \,, \tag{3.32}$$

$$\nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^{*},h) = \begin{cases} -\,(\boldsymbol{r}-\boldsymbol{r}^{*})\,\dfrac{k_k}{h^{D+2}}\,5.0\left(1.0-0.5\dfrac{\|\boldsymbol{r}-\boldsymbol{r}^{*}\|}{h}\right)^3 & q \leq 2 \\ 0 & q > 2 \end{cases} \,, \tag{3.33}$$

with

$$
\begin{aligned}
1\text{D} \rightarrow k_k &= \frac{3}{4}\,, \\
2\text{D} \rightarrow k_k &= \frac{7}{4\pi}\,, \\
3\text{D} \rightarrow k_k &= \frac{21}{16\pi}\,.
\end{aligned}
$$



(a) $M(q)$ and its derivative (b) $F(q)$

Figure 3.1: C2 Wendland (WC2) kernel function used

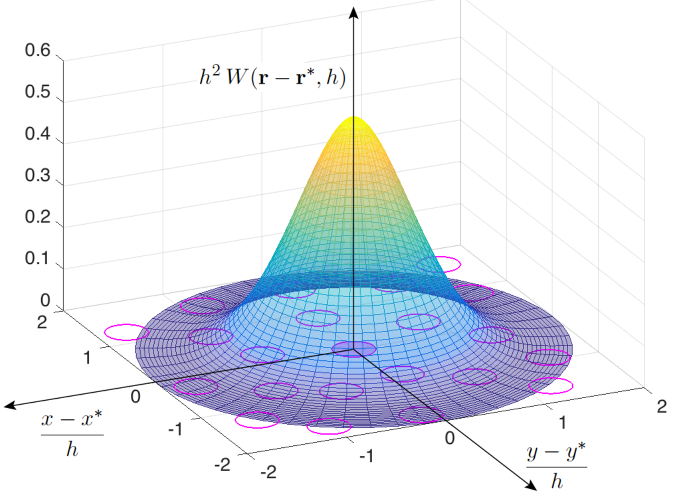


Figure 3.2: WC2 $W(\boldsymbol{r}-\boldsymbol{r}^*,h)$ in 2D centered at $\boldsymbol{r}$

3.1.3 Diffusive terms (δ-SPH correction)

In order to avoid the excessive pressure oscillations, Molteni and Colagrossi (2009) explored adding an additional diffusive term in the smoothed continuity equation. Antuono et al. (2010,

2012) expanded this concept, formally establishing the δ-SPH scheme that we use in the SPH sub-domain. The continuity Eq. (2.8) is written in its SPH form as

$$< \frac{\mathrm{d}\rho}{\mathrm{d}t} >_a \; = \; -\rho_a \; < \nabla \cdot \boldsymbol{u} >_a \; + \; \left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} \; . \tag{3.34}$$

with the new artificial diffusion term written in its general form as a function of a variable $\boldsymbol{\psi}$ such that

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} \; = \delta\, c \sum_b \bar{h}_{ab} \frac{m_b}{\rho_b} \boldsymbol{\psi}_{ab} \cdot \nabla_a W_{ab} \, . \tag{3.35}$$

where δ is a parameter used to adjust the intensity of this diffusion coefficient. Like the artificial viscosity, multiple formulations for $\boldsymbol{\psi}$ exist in the literature. We will adopt in this work the formulation proposed by Cercos-Pita (2016) and Cercos-Pita et al. (2016) based on its good consistency and conservation properties, and the fact that it doesn't require tuning parameters:

$$\delta\, c\, \bar{h}_{ab}\, \boldsymbol{\psi}_{ab} \; = \; \frac{\Delta t\, \rho_a}{\rho_0} \left((p_b - p_a) \frac{\boldsymbol{r}_{ab}}{||\boldsymbol{r}_{ab}||^2} + f_L(p_a, p_b) \right) , \tag{3.36}$$

where Δt is the time step in the numerical scheme and f_L is a correction coefficient when close to the boundaries. Since in the coupled model presented no SPH particle interacts with a boundary, we can drop f_L from our formulation. Furthermore, we include an additional term to remove the expected pressure difference from the body forces $\Delta p_{\boldsymbol{g}}$, *i.e.* its hydrostatic component:

$$\Delta p_{\boldsymbol{g},\, ab} = \bar{\rho}_{ab}\, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} \, . \tag{3.37}$$

Including Eqs. (3.36)-(3.37) into Eq. (3.35), we arrive at:

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} \; = \sum_b \frac{m_b}{\rho_b} \frac{\Delta t\, \rho_a}{\rho_0} \left(p_b + \bar{\rho}_{ab}\, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} - p_a\right) \frac{\boldsymbol{r}_{ab}}{||\boldsymbol{r}_{ab}||^2} \cdot \nabla_a W_{ab} \, . \tag{3.38}$$

Based on the expression of ∇W from Eq. (3.22), we can further simplify this equation:

$$F_{ab} \; = \; F\left(\frac{||\boldsymbol{r}_{ab}||}{\bar{h}_{ab}}\right) \quad \rightarrow \quad \nabla_a W_{ab} \; = \; \boldsymbol{r}_{ab}\, F_{ab} \, , \tag{3.39}$$

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} \; = \sum_b \frac{m_b}{\rho_b} \frac{\Delta t\, \rho_a}{\rho_0} \left(p_b + \bar{\rho}_{ab}\, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} - p_a\right) F_{ab} \, . \tag{3.40}$$

Physically, Eq. (3.40) opposes the natural density change from the pressure jump[1], effectively suppressing the induced sound wave and smoothing the density field. The main caveat of this formulation is its lack of symmetry, *i.e.* the effect from particle a on particle b is different than that from b on a. However, since $\rho_a \approx \rho_b \approx \rho_0$, this differences can be negligible (Cercos-Pita et al., 2016).

---

[1] $p_b > p_a \implies \rho_b > \rho_a \rightarrow \rho_a \uparrow, \rho_b \downarrow$ to reach equilibrium. However, since $F_{ab} < 0 \implies \left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} < 0$

3.1.4 δ-SPH SCHEME USED

The δ-SPH formulation is used to approximate the spatial derivatives in the Navier–Stokes governing Eqs. (2.8)-(2.9). As a result, we obtain a set of ODEs used in the SPH sub-domain to compute the numerical solution.

The only special treatment is applied to the pressure gradient, following the common practice in SPH. Rather than the expansion used for the general gradient formulation in Eq. (3.9), the following approach is used here:

$$\frac{\nabla p}{\rho} = \nabla\left(\frac{p}{\rho}\right) + \frac{p}{\rho^2}\nabla\rho\,, \tag{3.41}$$

$$< \frac{\nabla p}{\rho} >_a = \frac{\nabla p_a}{\rho_a} = \sum_b m_b\left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right)\nabla_a W_{ab}\,, \tag{3.42}$$

where the direct smoothing relation from Eq. (3.8) has been used to smooth the RHS terms of Eq. (3.41). The resulting approximated pressure gradient does not vanish for constant fields, but it conserves exactly linear and angular momentum as the force on a due to b is equal and opposite to the force on b due to a. (Monaghan, 2005b; Colagrossi et al., 2009).

After we take into consideration all the expressions derived in the previous sections, the evolution equations of the final δ-SPH scheme used, for the a-th particle, read:

$$\begin{cases} \dfrac{\mathrm{d}\rho_a}{\mathrm{d}t} = -\rho_a\nabla\cdot\boldsymbol{u}_a + \displaystyle\sum_b \frac{m_b}{\rho_b}\frac{\Delta t\,\rho_a}{\rho_0}\left(p_b + \bar{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right)F_{ab}\,, \\ \dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = \boldsymbol{g}_a - \dfrac{1}{\rho_a}\displaystyle\sum_b m_b\left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right)\boldsymbol{r}_{ab}\,F_{ab} \\ \qquad\qquad - \displaystyle\sum_b\left(\nu\,2(D+2) + \alpha\,\bar{h}_{ab}\,c\right)m_b\frac{\bar{\rho}_{ab}}{\rho_a\,\rho_b}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\boldsymbol{r}_{ab}\,F_{ab}\,, \\ \dfrac{\mathrm{d}\boldsymbol{r}_a}{\mathrm{d}t} = \boldsymbol{u}_a\,, \end{cases} \tag{3.43}$$

with

$$\nu \neq 0 \implies \alpha = 0\,,$$

$$-\rho_a\nabla\cdot\boldsymbol{u}_a = \begin{cases} \displaystyle\sum_b m_b\,\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}\,F_{ab} & \rho_{0,a} = \rho_{0,b}\,, \\ \rho_a\displaystyle\sum_b\frac{m_b}{\rho_b}\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}\,F_{ab} & \rho_{0,a} \neq \rho_{0,b}\,, \end{cases} \tag{3.44}$$

$$p_a = \begin{cases} c^2\,(\rho_a - \rho_0) + p_0 & \gamma = 1\,, \\ \dfrac{c^2\rho_0}{\gamma}\left[\left(\dfrac{\rho_a}{\rho_0}\right)^{\gamma} - 1\right] + p_0 & \gamma \neq 1\,, \end{cases} \tag{3.45}$$

$$\left(\frac{h_a}{h_0}\right)^D = \frac{\rho_0}{\rho_a}\,. \tag{3.46}$$

By default we use Eq. (3.13) to compute the divergence of the velocity, since densities cancel leading to a simpler and more robust formulation. However, when two or more fluids are present with large density ratios,[2] Eq. (3.13) is more accurate since it involves explicitly the density in the summation (Colagrossi, 2005). The artificial viscosity component is only applied in inviscid flows, as otherwise it will just unnecessarily modify the real fluid viscosity.

ρ_0 and ν are set based on the fluid properties. p_0 is a background pressure, useful for avoiding possible negative pressures. It is set to zero for free-surface flows. c is selected following the guidelines from Sec. 2.3, and h_0 is the reference smoothing length. Defining the particle size as

$$\Delta x_a = (\mathcal{V}_a)^{1/D} = \left(\frac{m_a}{\rho_a}\right)^{1/D} \tag{3.47}$$

for particle a, we specify the reference smoothing length as a multiple of the particle size at the reference pressure to capture an adequate number of neighbors. In the present work:

$$h_0 = 2\,\Delta x_{a,0} = 2\left(\frac{m_a}{\rho_0}\right)^{1/D}\,, \tag{3.48}$$

that corresponds to an average number of particles in the kernel support of about 50. The smoothing length may vary slightly from the reference value h_0 along the simulation for each particle so that the number of neighbors remains approximately constant.

---

[2] Within one fluid domain, no large density changes are expected based on the weakly compressible assumption as equation of state. Therefore, the large jumps in density between neighboring particles should only be caused by different fluid types

3.2 VPH SCHEME

3.2.1 GENERAL

In a Voronoi tessellation (dual construction of a Delaunay triangulation), a polyhedral volume is assigned to each particle, which encompasses the space closer to that particle than to any other. By itself, it is a pure geometrical construction based only on the location of the particles. Various optimized algorithms for its generation are already available, as presented in Sec. 1.3.3. In this work, the open-source C library *qHull* is used for its good performance and scalability to 3D and higher dimensions (Barber et al., 1996). As graphically shown in Fig. 3.3, the resulting Voronoi tessellation for a D-dimensional space is characterized by:

Nodes, given by their set of coordinates $\mathcal{X}_V$.

Faces, where each face represent the contact surface between two neighbor particles. It is orthogonal to the vector connecting both particles, and placed half-way between them. Each Voronoi face is a simplex, *i.e.* a D-dimensional triangle, and therefore consists always of a set of D nodes. They are given by

- Face connectivity list, specifying the nodes linked to each face, $\mathcal{F}_V$.
- Face particles, *i.e*, set of 2 particles linked to each face, $\mathcal{F}_P$.

Cells, given by the set of faces that delimit the polyhedral volume associated to each cell, C_F. Note that each particle cell can have different number of associated faces.

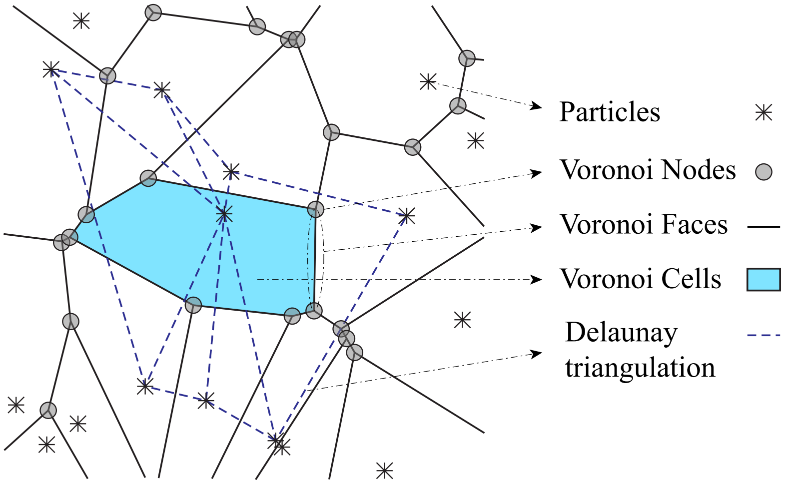


Figure 3.3: Voronoi tessellation

Based on this partition of the fluid domain, Serrano and Español (2001) devised a methodology to calculate the spatial derivatives, with many similarities to SPH. Later, Hess and Springel (2010) applied this concept to solve for fluid flows, formally establishing the basis for Voronoi particle hydrodynamics (VPH). The method starting point is defining a smoothing function χ linked to a given cell a (Flekkøy et al., 2000), such that

$$\chi_a(\boldsymbol{r}_a - \boldsymbol{r}^*, h) = \frac{\exp\left(-\frac{\|\boldsymbol{r}_a - \boldsymbol{r}^*\|^2}{2\,h^2}\right)}{\sum_b \exp\left(-\frac{\|\boldsymbol{r}_b - \boldsymbol{r}^*\|^2}{2\,h^2}\right)} \,. \tag{3.49}$$

Taking its limit, we arrive at the Voronoi characteristic function:

$$\lim_{h\to 0} \chi_a = \prod_b H\left(\|\boldsymbol{r}_a - \boldsymbol{r}^*\| - \|\boldsymbol{r}_b - \boldsymbol{r}^*\|\right) \tag{3.50}$$

where H is the Heaviside step function in this case. Similar to the Dirac delta in SPH where we were looking at a single point, in this case this function becomes one within the Voronoi cell a and zero everywhere else. Following a similar approach to SPH, we can use this Voronoi characteristic function to determine an approximation of the differential operators. The formal derivation is not trivial, and we refer the reader to the work by Serrano and Español (2001) and Serrano (2002).

Conceptually, we use the cell information as a unique domain of integration for each particle, so we can approximate any field value at a particle a location as a volume average on the given cell, $[f]_a$, such that

$$[f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} f(\boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* \,, \tag{3.51}$$

where $\mathcal{V}_a$ is determined from the volume of the Voronoi cell (Appendix D.1). Unlike smoothing, this approach does not provide a continuous field based on the discrete set of particles. The particle approximation in this case turns into approximating the value of $f(\boldsymbol{r})$ as constant within the cell:

$$f(\boldsymbol{r}) = f(\boldsymbol{r}_a) \quad \forall \boldsymbol{r} \in \mathcal{V}_a \,. \tag{3.52}$$

Equation (3.51) becomes trivial for f, but provides a starting point to evaluate the derivatives.

$$[\nabla f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} \nabla f(\boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* \,. \tag{3.53}$$

For consistency, we will follow the same simplified notation convention of SPH (Sec. 3.1.1), with $f_a = f(\boldsymbol{r}_a) = [f(\boldsymbol{r})]_a$ and $\nabla f_a = \nabla f(\boldsymbol{r}_a) = [\nabla f(\boldsymbol{r})]_a$ in this case.

If we use the divergence theorem, we'll arrive at $[\nabla f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \int_{\mathcal{S}_a} f(\boldsymbol{r}^*)\, \mathbf{d}\boldsymbol{\mathcal{S}}^*$, where $\mathcal{S}$ indicates the face surfaces that enclose the cell. This approach however leaves us with the problem of evaluating f at the faces, where its value is discontinuous since it is shared by two neighboring cells. Alternatively, if we start by applying the divergence theorem on $(\mathbf{1} \cdot \boldsymbol{r})\, \nabla f(\boldsymbol{r})$ following Hess and Springel (2010), where $\mathbf{1}$ indicates an arbitrary constant unit vector, we show in Appendix B how the following equality is valid:

$$\int_{\mathcal{V}_a} \nabla f(\boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* = \int_{\mathcal{S}_a} \boldsymbol{r}^* \left(\nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^*\right) - \int_{\mathcal{V}_a} \boldsymbol{r}^*\, \nabla^2 f(\boldsymbol{r}^*) \mathrm{d}\mathcal{V}^* \,. \tag{3.54}$$

We can subdivide the integral over $\mathcal{S}_a$ as a summation over each individual cell face $\mathcal{S}_{ab}$, whose normal vector is precisely the unit vector pointing from particle a to particle b (see Fig. 3.4).

$$\mathbf{d}\boldsymbol{\mathcal{S}}_{ab} = \boldsymbol{e}_{ba}\,\mathrm{d}\mathcal{S}_{ab}\,, \tag{3.55}$$

$$\boldsymbol{e}_{ba} = \frac{\boldsymbol{r}_b - \boldsymbol{r}_a}{\|\boldsymbol{r}_b - \boldsymbol{r}_a\|} = -\boldsymbol{e}_{ab}\,. \tag{3.56}$$

Furthermore, if the field values are sufficiently smooth we can assume a linear approximation for the gradient at each cell face as

$$\nabla f(\boldsymbol{r}^{*})|_{\mathcal{S}_{ab}} \approx \frac{f_b - f_a}{\|\boldsymbol{r}_b - \boldsymbol{r}_a\|}\,\boldsymbol{e}_{ba} = \frac{f_{ab}}{\|\boldsymbol{r}_{ab}\|}\,\boldsymbol{e}_{ab}\,. \tag{3.57}$$

Under this smooth fields approximations, and if the particles fall close to the centroid of the cells, the volume integral on the RHS of Eq. (3.54) can be neglected. Based on these assumptions, Appendix B summarizes the derivation presented by Hess and Springel (2010), showing how we arrive at the following approximation for the gradient operator:

$$\nabla f_a = \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\, f_{ab}\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right], \tag{3.58}$$

where the neighboring particles b in VPH are defined as those with common cell faces with particle a, A_{ab} is the area of cell face between particles a and b (Appendix D.2), and $\boldsymbol{c}_{ab}$ is a vector going from the midpoint between a and b to the centroid of their associated face, as shown in Fig. 3.4.

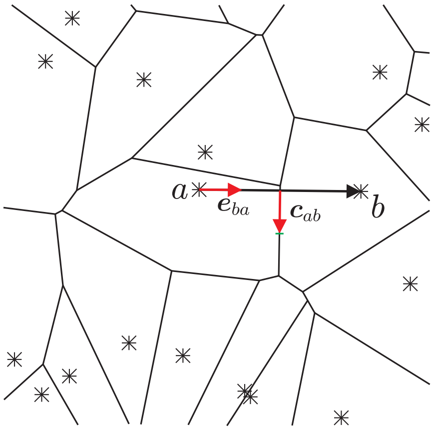


Figure 3.4: Section of a Voronoi diagram, for a set of particles marked with asterisks, showing the vectors $\boldsymbol{e}_{ba}$ and $\boldsymbol{c}_{ab}$.

It is straightforward to extend this formulation to compute the divergence of a vector field:

$$\nabla\cdot\boldsymbol{f}_a = \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\,\boldsymbol{f}_{ab}\cdot\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]. \tag{3.59}$$

We must also highlight that, for linear fields, these first order differential operators are exact, independently of the particle locations (Serrano and Español, 2001), which is a significant improvement from the SPH approach.

Like in SPH, the Laplacian operator of a vector field is approximated following an alternative approach. Starting with the general volume averaged expression and using directly the divergence theorem in this case,

$$\left[\nabla^2 \boldsymbol{f}(\boldsymbol{r})\right]_a = \frac{1}{\mathcal{V}_a}\int_{\mathcal{V}_a} \nabla^2 \boldsymbol{f}\, \mathrm{d}\mathcal{V}^* = \frac{1}{\mathcal{V}_a}\int_{\mathcal{S}_a} \nabla \boldsymbol{f} \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* = \frac{1}{\mathcal{V}_a}\sum_{b\neq a}\int_{\mathcal{S}_{ab}} \nabla \boldsymbol{f} \cdot \boldsymbol{e}_{ba}\, \mathrm{d}\mathcal{S}^* \,. \tag{3.60}$$

We can interpret the dot product inside the integral as the directional derivative of f. Same as in the derivation for the gradient, we can use the linear approximation for the gradient in Eq. (3.57) so

$$\nabla \boldsymbol{f} \cdot \boldsymbol{e}_{ba} = \frac{\boldsymbol{f}_{ab}}{\|\boldsymbol{r}_{ab}\|}\boldsymbol{e}_{ab} \cdot \boldsymbol{e}_{ba} = -\frac{\boldsymbol{f}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,, \tag{3.61}$$

which leads to the following estimate for the Laplacian operator:

$$\nabla^2 \boldsymbol{f}_a = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\boldsymbol{f}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,. \tag{3.62}$$

The order of this formula is not clearly established in literature. Serrano (2006) used it to simulate a shear stationary flow, showing that the schemes display extra dissipation when particles are in disordered configurations while it renders accurate results for orderly ones. For an extensive discussion on the properties of the Laplacian operator in Voronoi and Delaunay tessellations, the reader is referred to Duque et al. (2017) and references therein.

3.2.2 ARTIFICIAL VISCOSITY

Same as in the SPH scheme, an artificial viscosity term is necessary for inviscid problems to achieve stability. We showed in Appendix A (Eq. (A.12)) how the AV in the SPH formulation can be expressed in the continuum form as

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV} \approx \frac{1}{2(D+2)}\,\alpha\, h\, c\,\left[\nabla^2 \boldsymbol{u} + 2\nabla(\nabla\cdot\boldsymbol{u})\right].$$

The component linked to $\nabla\cdot\boldsymbol{u}$ in weakly compressible flows is usually negligible. However, under certain flows $\nabla^2\boldsymbol{u}\approx 0$, so we will keep both components for the artificial viscosity formulation. Appendix C discusses the stability of a simple uni-dimensional configuration, showing why this numerical damping term is needed and demonstrating its chosen order of magnitude (αhc).

If we follow the general VPH approach, the approximation of $\nabla(\nabla\cdot\boldsymbol{u})$ would lead to double summations that are computationally expensive[3]. Same as with the Laplacian operator, we

---

[3] Combining Eqs. (3.58) and (3.59), we would arrive at

$$\begin{aligned}\nabla(\nabla\cdot\boldsymbol{f}_a) &= \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\,(\nabla\cdot f_a - \nabla\cdot f_b)\cdot\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]\\ &= \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\left(\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\,\boldsymbol{f}_{ab}\cdot\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right] - \frac{1}{\mathcal{V}_b}\sum_{c\neq b} A_{bc}\,\boldsymbol{f}_{bc}\cdot\left[\frac{\boldsymbol{e}_{bc}}{2} - \frac{\boldsymbol{c}_{bc}}{\|\boldsymbol{r}_{bc}\|}\right]\right)\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]\end{aligned}$$

where c goes through all the neighbors of each neighboring particle b

approach this component from a simpler perspective, leading to an alternative expression much easier to compute. Starting again from the volume averaged expression, we can apply directly the divergence theorem, leading to

$$[\nabla(\nabla\cdot\boldsymbol{u})]_a = \frac{1}{\mathcal{V}_a}\int_{\mathcal{V}_a}\nabla(\nabla\cdot\boldsymbol{u})\,\mathrm{d}\mathcal{V}^* = \frac{1}{\mathcal{V}_a}\int_{\mathcal{S}_a}(\nabla\cdot\boldsymbol{u})\,\mathbf{d}\mathcal{S}^* = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a}\int_{\mathcal{S}_{ab}}(\nabla\cdot\boldsymbol{u})\,\boldsymbol{e}_{ab}\,\mathrm{d}\mathcal{S}^* \,. \tag{3.63}$$

Looking at the physical meaning of the divergence of the velocity, from the continuity Eq. (2.8) we see how it represents the change of volume of the cell.

$$\nabla\cdot\boldsymbol{u} = -\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\frac{\mathcal{V}}{m}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{m}{\mathcal{V}}\right) = \frac{1}{\mathcal{V}}\frac{\mathrm{d}\mathcal{V}}{\mathrm{d}t} \,. \tag{3.64}$$

Since the divergence is evaluated on the faces in this case, we can interpret it as how each face contributes to the overall change of volume, *i.e.* whether the face moves towards the particle or away from it. Each face is placed halfway between the particle a and its neighbor b in a Voronoi diagram, allowing us to determine its relative velocity to particle a as,

$$\boldsymbol{u}_{\text{face}} = \frac{\boldsymbol{u}_b + \boldsymbol{u}_a}{2} - \boldsymbol{u}_a = \frac{\boldsymbol{u}_b - \boldsymbol{u}_a}{2} \,. \tag{3.65}$$

We can use Eq. (3.65) to approximate the change of cell volume due to the face movement as

$$\mathcal{V}_{\text{face}} \approx A_{ab}\frac{\|\boldsymbol{r}_b - \boldsymbol{r}_a\|}{2} \,, \tag{3.66}$$

$$\left.\frac{\mathrm{d}\mathcal{V}}{dt}\right|_{\text{face}} \approx \frac{A_{ab}\,\|\mathbf{d}\boldsymbol{r}\|_{ba}}{dt} = A_{ab}\frac{\mathrm{d}\boldsymbol{r}\cdot\boldsymbol{e}_{ba}}{dt} = A_{ab}\,\boldsymbol{u}_{\text{face}}\cdot\boldsymbol{e}_{ba} = A_{ab}\frac{\boldsymbol{u}_{ba}\cdot\boldsymbol{e}_{ba}}{2} \,. \tag{3.67}$$

Substituting it into Eq. (3.64), we can approximate the divergence on the face as

$$(\nabla\cdot\boldsymbol{u})_{\text{face}} \approx \frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|} \,, \tag{3.68}$$

which substituted in Eq. (3.63) yields the VPH approximation for the gradient of the divergence:

$$[\nabla(\nabla\cdot\boldsymbol{u})]_a = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a}A_{ab}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|}\boldsymbol{e}_{ab} \,. \tag{3.69}$$

The last step is using the derived expressions for the differential operators in Eq. (A.12) for the artificial viscosity to determine its form under the VPH methodology

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV,\,VPH} = \frac{1}{2(D+2)}\,\alpha\,h\,c\left[-\frac{1}{\mathcal{V}_a}\sum_{b\neq a}A_{ab}\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} - 2\frac{1}{\mathcal{V}_a}\sum_{b\neq a}A_{ab}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|}\boldsymbol{e}_{ab}\right] , \tag{3.70}$$

or in a more compact form, and considering that h can vary

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV,\,VPH} = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a}A_{ab}\frac{\alpha\,\overline{h}_{ab}\,c}{2(D+2)}\left[\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} + 2\left(\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|}\cdot\boldsymbol{e}_{ab}\right)\boldsymbol{e}_{ab}\right] . \tag{3.71}$$

Strictly, the smoothing length is not present in the VPH scheme and it should be replaced by the particle size $\Delta x_a = (m_a/\rho_a)^{1/D}$, as done in Appendix C. However, h is kept in order to be consistent with the δ-SPH formulation and facilitate the coupling later.

3.2.3 DIFFUSIVE TERMS

The density can be estimated from the Voronoi cell. However, it is more beneficial to use the continuity equation to evolve it in time, so we can include a diffusive term to reduce the high-order pressure oscillations as discussed in SPH (Sec. 3.1.3). The same formulation used in the δ-SPH scheme can be extended to the VPH, although no previous reference in the literature was found. Comparing the SPH and VPH formulations, we can extract the following analogy:

$$\sum_b \frac{m_b}{\rho_b} (\dots) \nabla_a W_{ab} \quad \leftrightarrow \quad \frac{1}{\mathcal{V}_a} \sum_{b \neq a} A_{ab} (\dots) \left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right], \tag{3.72}$$

which applied to diffusive term used in SPH (Eq. (3.38)) leads to

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-VPH} = \frac{1}{\mathcal{V}_a} \sum_{b \neq a} A_{ab} \frac{\Delta t \, \rho_a}{\rho_0} \left(p_b + \overline{\rho}_{ab} \, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} - p_a\right) \frac{\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2} \cdot \left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right]. \tag{3.73}$$

Noting that $\boldsymbol{e}_{ab} = \frac{\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|}$ is orthogonal to $\boldsymbol{c}_{ab}$ (so $\boldsymbol{r}_{ab} \cdot \boldsymbol{c}_{ab} = 0$), we can simplify this equation and arrive at the final expression for the diffusive term used:

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-VPH} = \frac{1}{\mathcal{V}_a} \sum_{b \neq a} A_{ab} \frac{\Delta t \, \rho_a}{\rho_0} \frac{\left(p_b + \overline{\rho}_{ab} \, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} - p_a\right)}{2 \, \|\boldsymbol{r}_{ab}\|}. \tag{3.74}$$

As a final remark, we should point out that since density is evolved in time unlike the volume that is computed from the Voronoi diagram, both values are not related. We can use the cell volume from the tessellation to determine a Voronoi density estimate, $\check{\rho}$, but it will not match the evolved density ρ.

$$\rho_a \neq \check{\rho}_a = \frac{m_a}{\mathcal{V}_a}. \tag{3.75}$$

3.2.4 δ-VPH SCHEME USED

Based on the previous expressions, we can again approximate the spatial derivatives in the Navier–Stokes governing Eqs. (2.8)-(2.9), reducing them to a set of ODEs used in the Voronoi sub-domain to evolve the flow variables.

Like in SPH, the pressure gradient component receives special treatment to ensure that it is pairwise antisymmetric, *i.e.* the force on a due to b is equal in magnitude and opposite to the force on b due to a, and therefore conserve angular and linear momentum. Under the VPH this is fairly simple, since a close domain always satisfies

$$\sum_{b \neq a} A_{ab} \boldsymbol{e}_{ab} = 0. \tag{3.76}$$

By subtracting $p_a \sum_{b \neq a} A_{ab} \, \boldsymbol{e}_{ab}$ from Eq. (3.58), we arrive at the desired expression for the pressure gradient used (Serrano and Español, 2001; Hess and Springel, 2010),

$$\nabla p_a = -\frac{1}{\mathcal{V}_a} \sum_{b \neq a} A_{ab} \left[(p_a + p_b) \frac{\boldsymbol{e}_{ab}}{2} + (p_a - p_b) \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right], \tag{3.77}$$

Finally, substituting the definitions presented for the differential operators, we obtain the evolution equations of the δ-VPH scheme proposed, which read for the a-th particle

$$\left\{\begin{aligned} \frac{\mathrm{d}\rho_a}{\mathrm{d}t} &= -\frac{\rho_a}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\,\boldsymbol{u}_{ab}\cdot\left[\frac{\boldsymbol{e}_{ab}}{2}-\frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]+\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\,\frac{\Delta t\,\rho_a}{\rho_0}\,\frac{(p_b+\bar{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab}-p_a)}{2\,\|\boldsymbol{r}_{ab}\|}\,,\\ \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} &= \boldsymbol{g}_a+\frac{1}{\rho_a\,\mathcal{V}_a}\sum_{b\neq a} A_{ab}\left[(p_a+p_b)\,\frac{\boldsymbol{e}_{ab}}{2}+(p_a-p_b)\frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]-\frac{\nu}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|}\\ &\qquad -\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\alpha\,\bar{h}_{ab}\,c}{2(D+2)}\left[\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|}+2\left(\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|}\cdot\boldsymbol{e}_{ab}\right)\boldsymbol{e}_{ab}\right],\\ \frac{\mathrm{d}\boldsymbol{r}_a}{\mathrm{d}t} &= \boldsymbol{u}_a\,, \end{aligned}\right. \tag{3.78}$$

with $\nu \neq 0 \implies \alpha = 0$, and using the same equation of state than in δ-SPH,

$$p_a = \begin{cases} c^2\,(\rho_a-\rho_0)+p_0 & \gamma = 1\,,\\ \dfrac{c^2\rho_0}{\gamma}\left[\left(\dfrac{\rho_a}{\rho_0}\right)^{\gamma}-1\right]+p_0 & \gamma \neq 1\,. \end{cases}$$

$\mathcal{V}_a$ and A_{ab} are computed from the Voronoi tessellation (Appendix D).

3.3 Coupling

3.3.1 SPH & VPH Sub-domains

The SPH and VPH schemes described in the previous sections are very similar, and their coupling arises as a natural way of capturing the advantages of each one. VPH reduces the interaction between particles to the immediate neighbors, *i.e.* those with shared faces. This is beneficial to enforce the boundary conditions if we are able to map exactly the boundaries with cell faces, as discussed in Secs. 3.4.2 and 3.4.3. On the other hand, SPH manages efficiently free-surfaces with large deformations.

We define two sub-domains in the fluid based on the distance to the closest boundary, d, computed as explained in Appendix D.5. VPH is used in the sub-domain closer to the boundaries, while SPH is used away from the boundaries. There is some overlapping between the two sub-domains to allow for a smooth transition between the two methods, so we distinguish three separate zones as shown in the sketch presented in Fig. 3.5:

Voronoi : Zone next to the boundaries where only VPH is used.

Buffer : Overlapping zone between both sub-domains, where variables are computed using both VPH and SPH.

SPH : Zone away from the boundaries where only SPH is used.

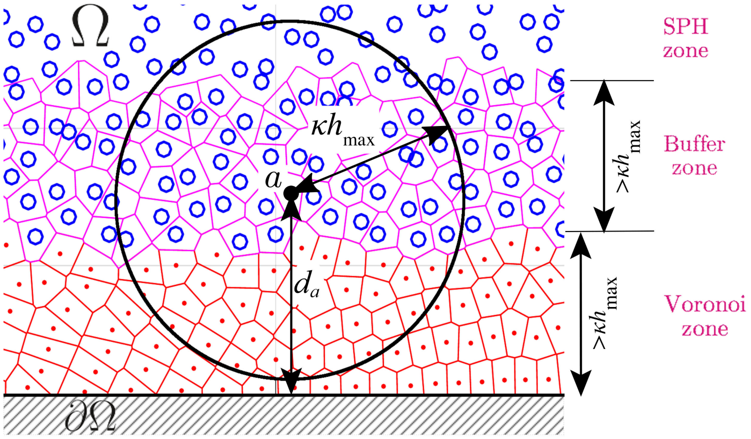


Figure 3.5: Sketch with definition of Voronoi, SPH and buffer zones.

The width of the Voronoi zone is such that all particles outside the Voronoi zone will be at distance to the closest solid boundary larger than $\kappa h_{\max}$, with κ being the factor that characterizes the support radius of the kernel[4] and $h_{\max}$ the largest smoothing length. Consequently, no particle belonging to the SPH sub-domain (buffer and SPH) will have any interaction with the solid boundaries, *i.e.* **we don't need to account for solid boundary conditions in the SPH formulation**.

---

[4] The C2 Wendland kernel with $\kappa = 2$ is used in this work (Sec. 3.1.2), so the limit is set to $2\,h_{\max}$

The thickness of the buffer zone, l_{buffer}, is also at least $\kappa h_{\max}$, so no particle in the SPH zone interacts with a particle in the Voronoi zone. However, it can be extended to achieve a smoother transition between methods, as discussed in Sec. 3.3.3. Finally, the SPH zone is simply the rest of the fluid domain.

At each time step, we will identify each particle as Voronoi, buffer, or SPH based on the zone where it lays, which depends on the position of the boundaries at that time step. We should stress that the particles themselves are just interpolation points where physical variables are evaluated (p, ρ, $\boldsymbol{u}$, $\mathrm{d}\rho/\mathrm{d}t$, etc.), and are not linked to any formulation. In this way, a "Voronoi particle" can become a "buffer particle" as it moves away from the boundary, and once it crosses the buffer become an "SPH particle", and vice versa.

3.3.2 FIELDS IN THE BUFFER ZONE

First, we should re-state that the value of the different field variables at the particle locations is unique, independent of the formulation used to compute them. In the buffer zone, however, fields are evaluated with both SPH and VPH so they need to be combined. Therefore, the value of a certain field A at a particle a when it is in the buffer is defined as:

$$A_a := \omega_a A_a^{SPH} + (1 - \omega_a) A_a^{Vor}, \tag{3.79}$$

where A_a^{SPH} is the field value obtained through the SPH interpolation, A_a^{Vor} is the field value obtained with the VPH interpolation, and ω is a weight defined so that the coupling is C^1. We must stress that this condition is a first best guess, inspired by the h^2 order of the SPH interpolation studied in detailed by Quinlan et al. (2006). However, other alternatives are of course possible. To this aim, the value A_a should tend to A_a^{Vor} as the particle gets close to the Voronoi zone and should tend to A_a^{SPH} as the particle gets close to the SPH zone. A factor q is defined using the distance d_a between the particle a and the closest solid wall:

$$q_a := \frac{d_a - \kappa h_{\max}}{l_{\text{buffer}}}, \tag{3.80}$$

and ω_a is the output of a third-grade polynomial in r_a (Fig. 3.6), which allows us to have the required C^1 class for the coupling scheme in Eq. (3.79):

$$\omega_a := q_a^2 (3 - 2\, q_a). \tag{3.81}$$

We have seen in the VPH expressions for the differential operators how we are looking at the interaction of a particle a with its neighbors b. Since each particle has a different weight, the remaining question is which one to use. Two possible approaches are considered:

1. Calculate the interaction with all neighbors and apply the weight in the final value of the derivative,

$$\nabla A_a = \omega_a \nabla A_a^{SPH} + (1 - \omega_a) \nabla A_a^{Vor}, \tag{3.82}$$

2. Average weights in the local interaction between particles a and b,

$$\nabla A_a = \sum_{b \neq a} \nabla A_{ab} = \sum_{b \neq a} \overline{\omega}_{ab} \nabla A_{b \to a}^{SPH} + (1 - \overline{\omega}_{ab}) \nabla A_{b \to a}^{Vor}, \tag{3.83}$$

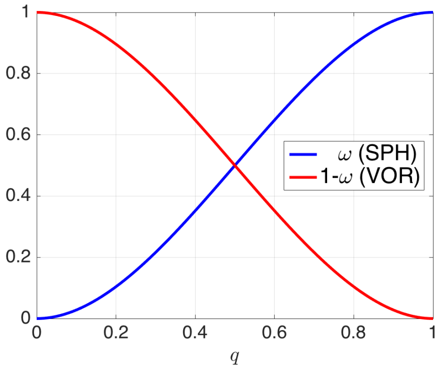


Figure 3.6: Coupling weight function ω as a function of its relative position across the buffer, q

The first option seems simpler, but is not conservative since the force on a due to b is different than on b due to a, even if the formulation is pair-wise antisymmetric in SPH and VPH. Looking at the pressure force as an example,

$$\begin{cases} \boldsymbol{f}_{p,\,b\to a} = m_a \dfrac{\nabla p_{ab}}{\rho_a} = \dfrac{m_a}{\rho_a}\left[\omega_a \nabla p^{SPH}_{b\to a} + (1-\omega_a)\nabla p^{Vor}_{b\to a}\right], \\ \boldsymbol{f}_{p,\,a\to b} = m_b \dfrac{\nabla p_{ba}}{\rho_b} = \dfrac{m_b}{\rho_b}\left[\omega_b \nabla p^{SPH}_{a\to b} + (1-\omega_b)\nabla p^{Vor}_{a\to b}\right]. \end{cases} \tag{3.84}$$

Both SPH and VPH formulations for ∇p_a (Eqs. (3.41)) and (3.77)) satisfy

$$m_a \frac{\nabla p^{SPH}_{b\to a}}{\rho_a} = -\frac{m_b}{\rho_b}\nabla p^{SPH}_{a\to b}, \tag{3.85}$$

$$m_a \frac{\nabla p^{Vor}_{b\to a}}{\rho_a} = -\frac{m_b}{\rho_b}\nabla p^{Vor}_{a\to b}. \tag{3.86}$$

In order to be conservative, $\boldsymbol{f}_{p,\,b\to a}$ and $\boldsymbol{f}_{p,\,a\to b}$ should cancel to conserve momentum, but due to the different weights it is not the case

$$\Delta \boldsymbol{f}_{ab} = \boldsymbol{f}_{p,\,b\to a} + \boldsymbol{f}_{p,\,a\to b} = \frac{m_a}{\rho_a}\,(\omega_a - \omega_b)\left(\nabla p^{SPH}_{b\to a} - \nabla p^{Vor}_{b\to a}\right) \neq 0. \tag{3.87}$$

The second option is fully pair-wise antisymmetric and therefore conservative overall. However, the individual components from SPH and VPH are unbalanced due to the different weight for each neighbors. For example, we can interpret this fact in SPH as a modification of the kernel which is no longer a symmetric function. Neighbors further from the boundary have a bigger impact with no physical reason. Likewise, the neighbor contribution in VPH is not just linked to the face area, becoming more relevant those neighbors closer to the boundary.

Ideally, these deviations should cancel each other. However, this is not the case due to the different characteristic lengths of VPH (Δx) and SPH (κh). Consequently, the number of neighbors involved is different, with different individual contributions in each method. We can quantify the order of this deviation, $\mathcal{O}(\Delta\nabla A_a)$, by subtracting both approaches

$$\begin{aligned}\mathcal{O}(\Delta\nabla A_a) &= \left|\left(\omega_a\,\nabla A_a^{SPH} + (1-\omega_a)\,\nabla A_a^{Vor}\right) - \left(\sum_{b:\mathrm{SPH}} \overline{\omega}_{ab}\,\nabla A_{b\to a}^{SPH} + \sum_{b:\mathrm{VPH}} (1-\overline{\omega}_{ab})\,\nabla A_{b\to a}^{Vor}\right)\right| \\ &= \left|\sum_{b:\mathrm{SPH}} \frac{\omega_a-\omega_b}{2}\,\nabla A_{b\to a}^{SPH} - \sum_{b:\mathrm{VPH}} \frac{\omega_a-\omega_b}{2}\,\nabla A_{b\to a}^{Vor}\right| .\end{aligned} \tag{3.88}$$

Based on the formulation used for the weight,

$$\max|\omega_a-\omega_b|_{\mathrm{Vor}} \le |\omega_a-\omega_b|_{\mathrm{SPH}} \;,$$

since the smoothing compact support is larger than the particle size so the possible maximum distances between SPH neighbors are equal or larger than in VPH ($\kappa h_{\max} > \Delta x$). We can therefore determine an upper bound to this deviation as

$$\mathcal{O}(\Delta\nabla A_a) \le \frac{\max|\omega_a-\omega_b|_{\mathrm{SPH}}}{2}\left[\sum_{b:\mathrm{SPH}} \nabla A_{b\to a}^{SPH} - \sum_{b:\mathrm{VPH}} \nabla A_{b\to a}^{Vor}\right] = \frac{\max|\omega_a-\omega_b|_{\mathrm{SPH}}}{2}\,\mathcal{O}\left((\kappa h)^2\right) , \tag{3.89}$$

where we have directly applied the known second order of the SPH and VPH approximations. If this approach is followed to compute the derivatives, we should limit this deviation to at least an order of magnitude lower than $\mathcal{O}\left((\kappa h)^2\right)$ to avoid loosing accuracy. We achieve this by adjusting the buffer length to limit the maximum weight variations, as discussed in Sec. 3.3.3 below, such that:

$$\mathcal{O}(\Delta\nabla A_a) \le 0.1\,\mathcal{O}\left((\kappa h)^2\right) \implies \frac{\max|\omega_a-\omega_b|_{\mathrm{SPH}}}{2} \le 0.1 \,. \tag{3.90}$$

3.3.3 LENGTH OF THE BUFFER ZONE

We can take the kernel compact support radius as an upper bound for the maximum difference in the distance to the boundary between a and b

$$\max|d_a-d_b| \le \kappa h_{\max} \,. \tag{3.91}$$

Without any loss of generality since the sub-indexes can be interchanged, we can assume $d_a > d_b \to \omega_a > \omega_b$, and remove the absolute values to determine an upper bound for the maximum expected weight differences

$$d_a \le d_b + \kappa h_{\max} \,. \tag{3.92}$$

Based on Eq. (3.80), we can express the relative position in the buffer of particle b as

$$q_a = \frac{d_a-\kappa h_{\max}}{l_{\mathrm{buffer}}} \le \frac{d_b-\kappa h_{\max}}{l_{\mathrm{buffer}}} + \frac{\kappa h_{\max}}{l_{\mathrm{buffer}}} = q_b + \frac{\kappa h_{\max}}{l_{\mathrm{buffer}}} \,. \tag{3.93}$$

Naming $\varepsilon = \frac{\kappa h_{\max}}{l_{\mathrm{buffer}}}$, we can use this expression to determine the maximum expected weight difference for the coupling scheme used as

$$\begin{aligned}\omega_a-\omega_b &\le (q_b+\varepsilon)^2(3-2(q_b+\varepsilon)) - q_b^2(3-2q_b)^2 \\ &\le \varepsilon^2(3-2\,\varepsilon) + 6q_b\varepsilon(1-\varepsilon-q_b) \,,\end{aligned}$$

with $\varepsilon \in [0, 1]$, $q_b \in [0, 1]$, and $q_b + \varepsilon \leq 1$ as shown in Fig. 3.7, and whose maximum for a given ε occurs at

$$\frac{\partial}{\partial q_b}(\omega_a - \omega_b) = -6\varepsilon\,(\varepsilon + 2q_b - 1) = 0 \implies q_b = \frac{1}{2}\,(1 - \varepsilon)\;, \tag{3.94}$$

$$\max|\omega_a - \omega_b|_{\mathrm{SPH}} \leq \frac{\varepsilon}{2}\left(3 - \varepsilon^2\right)\,. \tag{3.95}$$

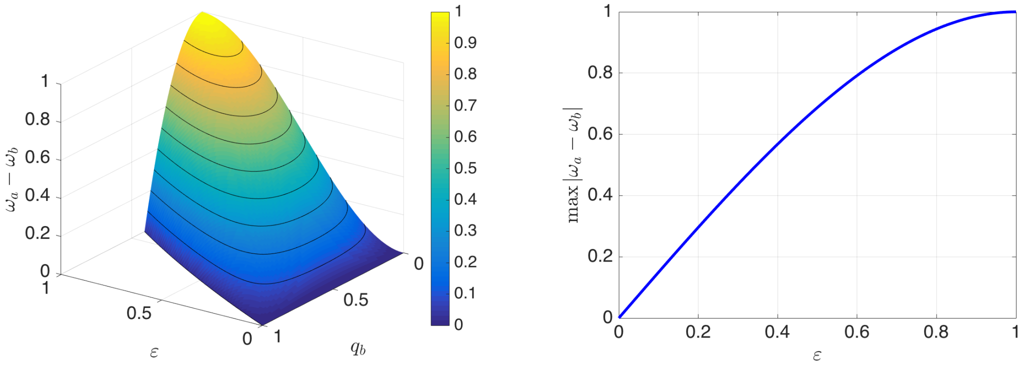


Figure 3.7: Upper bound of weight differences between neighbor particles.

We finally use this upper bound for $|\omega_a - \omega_b|$ in Eq. (3.90), to determine the minimum buffer length

$$\frac{\max|\omega_a - \omega_b|_{\mathrm{SPH}}}{2} = \frac{\varepsilon(3 - \varepsilon^2)}{4} \leq 0.1\,, \tag{3.96}$$

which, to the leading order, yields the following minimum buffer length[5],

$$\frac{3}{4}\varepsilon = \frac{3}{4}\,\frac{\kappa h_{\max}}{l_{\mathrm{buffer}}} < 0.1 \implies l_{\mathrm{buffer}} \geq 7.5\,\kappa h_{\max}\,. \tag{3.97}$$

Based on the previous considerations, the following practices have been followed in this work:

Compute derivatives independently if a non-conservative scheme is acceptable for the problem solved *(option 1)*. The buffer length is set to its minimum ,

$$l_{\mathrm{buffer}} = \kappa h_{\max}\,,$$

to minimize the area where forces between particles are not balanced, as well as the extra computational cost from calculating the field variables with both methods.

Average weights between neighbors to keep a conservative scheme *(option 2)*. The buffer length is set as

$$l_{\mathrm{buffer}} = 8\,\kappa h_{\max}$$

to maintain the order of each method as discussed above.

---

[5] Note that the expression obtained is based on the coupling scheme followed (Eq. (3.81)), and should be revised if a different scheme is used.

3.4 Boundary conditions (BCs)

3.4.1 Geometric definition requirements

- All surfaces bounding the fluid domain are discretized into a set of $D-1$ dimensional triangles. In this way, they are always flat surfaces, same as the Voronoi faces.
- The nodes that define each face are oriented so their normal vector points towards the fluid, as defined in Appendix D.3.
- No particle can interact with more than one boundary node. Therefore, an upper bound for the largest admissible particle size is established as half the length of the smallest boundary edge.

3.4.2 Convex solid boundaries

Under the coupled scheme, only Voronoi particles interact with solid boundaries. Consequently, the solid BCs need to be imposed only within the VPH formulation. The main challenge lies on achieving that the solid boundary becomes the actual boundary of well-defined Voronoi cells. The main topological limitation is that Voronoi cells are always convex when observed from the fluid particle. We must then distinguish between convex and concave adjacent boundary faces as explained in Appendix D.7, since only the convex ones can be overlapped with “pure” Voronoi cell faces. Concave adjacent faces require a different treatment, discussed in Sec. 3.4.3.

Voronoi diagrams are unbounded by definition, representing a partition of the complete spatial domain. Those particles physically in contact with the boundary will result in Voronoi cells that extend beyond the fluid domain, and need to be bounded. We can simply do this by noticing that in a symmetric configuration of particles, the Voronoi cells should be symmetric as well. Therefore, the symmetry plane should be mapped with cell faces since it cannot be crossed. Taking advantage of this fact, the boundary faces are converted into symmetry planes, and effectively become part of the tessellation, by creating a set of points that reflect the particles in the boundary nearby, as shown in Fig. 3.8.

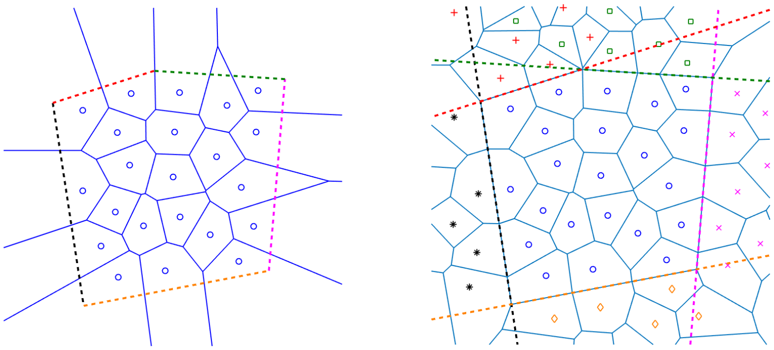

(a) Unbounded cells (b) Bounded cells

Figure 3.8: Voronoi diagrams in the presence of convex boundaries

Unlike ghost particles in SPH, these reflected points are used only to create the Voronoi tessellation, but do not play any role in the mathematical treatment of the flow fields. We’ll refer to them as boundary virtual points, and identify them with a superscript ^{B}. A priori, we cannot know which fluid particles are in contact with the boundary, so all particles within a distance threshold are reflected[6]. Each boundary virtual point is associated to one boundary face, so when it is identified as a neighbor we know the type of solid BC to be applied. In order to show how they are imposed, a sample particle a adjacent to a boundary, and its reflected boundary virtual point a^B will be used as an example[7], as displayed in Fig. 3.9.

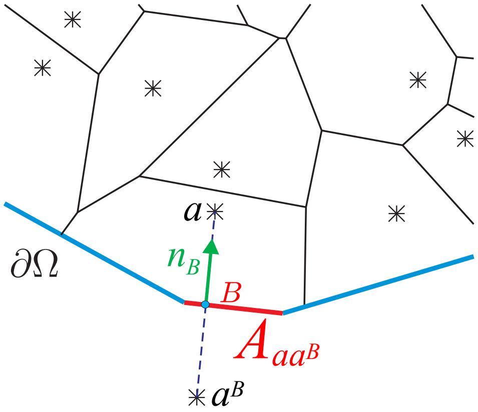


Figure 3.9: Section of a Voronoi diagram close to a solid boundary.

Once the solid boundaries become part of the perimeter of the Voronoi cells, the BCs from Sec. 2.1 can be enforced on the related terms in the VPH formulation (Sec. 3.2.4). This is done by expressing the required field values at a^B as functions of the known physical properties at the boundaries. We should recall that linear approximations were used to derive the VPH differential operators under the assumption of smooth fields (Sec. 3.2.1, Appendix B). Under the same assumption, we arrive at the following relations:

Boundary velocity, $\boldsymbol{u}_B$. Two alternatives can be applied (Sec. 2.1):

1. No-slip BC: the flow velocity tends to the boundary velocity as a fluid element gets closer to the boundary. To achieve this:

$$\boldsymbol{u}_{a^B} = 2\,\boldsymbol{u}_B - \boldsymbol{u}_a\,, \tag{3.98}$$

$$\boldsymbol{u}_{aa^B} = \boldsymbol{u}_a - \boldsymbol{u}_{a^B} = 2(\boldsymbol{u}_a - \boldsymbol{u}_B) = 2\,\boldsymbol{u}_{aB}\,. \tag{3.99}$$

2. Free slip BC: the normal velocity tends to the boundary normal velocity as a fluid element gets closer to the boundary. Eq. (3.98) is then projected on the boundary normal $\boldsymbol{n}_B = \boldsymbol{e}_{aa^B}$, while the tangential component, $\boldsymbol{u}_{a^B,t}$, remains unaltered:

$$\boldsymbol{u}_{a^B} \cdot \boldsymbol{n}_B = (2\,\boldsymbol{u}_B - \boldsymbol{u}_a) \cdot \boldsymbol{n}_B\,, \tag{3.100}$$

$$\boldsymbol{u}_{a^B,t} = \boldsymbol{u}_{a,t} = \boldsymbol{u}_a - (\boldsymbol{u}_a \cdot \boldsymbol{n}_B)\boldsymbol{n}_B\,. \tag{3.101}$$

---

[6] In this work, the condition used is $d_a < \Delta x_0$. Note that the distance to the boundaries needs also to be calculated to determine if it is a Voronoi, buffer, or SPH particle.

[7] Note that following the requirements in Sec. 3.4.1, the solid boundary is discretized as a polygonal line in 2D and no more than one boundary node can be covered by the cell.

$\boldsymbol{u}_{aa^B}$ should then be aligned with the normal vector, so

$$\boldsymbol{u}_{aa^B} = (2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B)\,\boldsymbol{n}_B \implies \begin{cases} \boldsymbol{u}_{aa^B}\cdot\boldsymbol{e}_{aa^B} = 2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B \\ \boldsymbol{u}_{aa^B}\cdot\boldsymbol{c}_{aa^B} = 0 \end{cases} \tag{3.102}$$

Boundary pressure, p_B. The pressure gradient is obtained by projecting the momentum equation on the boundary normal to enforce the following Neumann condition (Marrone et al., 2011):

$$\frac{\partial p}{\partial\boldsymbol{n}_B} = \rho\left[\boldsymbol{g}\cdot\boldsymbol{n}_B - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B + \nu\,\nabla^2\boldsymbol{u}|_B\cdot\boldsymbol{n}_B\right], \tag{3.103}$$

with the last term being generally negligible[8]. Noting that $\boldsymbol{n}_B$ is oriented towards the fluid, and using the centroid of the cell face as the representative point to account for hydrostatic correction, we approximate the constant p_B at the face by linear extrapolation as:

$$p_B = p_a - \frac{\partial p}{\partial\boldsymbol{n}_B}\frac{\|\boldsymbol{r}_{aa^B}\|}{2} + \rho_a\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B} = p_a - \rho_a\,\boldsymbol{g}\cdot\left(\frac{\boldsymbol{r}_{aa^B}}{2} - \boldsymbol{c}_{aa^B}\right) + \rho_a\frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\frac{\boldsymbol{r}_{aa^B}}{2}, \tag{3.104}$$

which leads to express the pressure at the boundary virtual point as

$$p_{a^B} = 2p_B - p_a = p_a - \rho_a\left[\left(\boldsymbol{g} - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right]. \tag{3.105}$$

Using these relations on the corresponding elements in the VPH evolution equations, given in Eq. (3.78), we arrive at the following contributions from the solid boundaries to the time derivatives:

1. No-slip BC, $\boldsymbol{u}_{aa^B} = 2(\boldsymbol{u}_a - \boldsymbol{u}_B) = 2\,\boldsymbol{u}_{aB}$

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = \overbrace{-\dfrac{\rho_a}{\mathcal{V}_a}A_{aa^B}\,\boldsymbol{u}_{aa^B}\cdot\left(\dfrac{\boldsymbol{n}_B}{2} - \dfrac{\boldsymbol{c}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\right)}^{\rho\nabla\cdot\boldsymbol{u}} + \overbrace{\dfrac{1}{\mathcal{V}_a}A_{aa^B}\,\dfrac{\Delta t\,\rho_a}{\rho_0}\,\dfrac{\rho_a}{2}\,\dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B}^{\text{diffusive term, }\delta\text{−VPH}}, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = \overbrace{\dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aa^B}\left[\rho_a\left(\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right)\left(\dfrac{\boldsymbol{n}_B}{2} - \dfrac{\boldsymbol{c}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\right) + p_a\,\boldsymbol{n}_B\right]}^{-\frac{\nabla p}{\rho}} \\ \qquad\qquad \underbrace{-\dfrac{\nu}{\mathcal{V}_a}A_{aa^B}\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}}_{\nu\nabla^2\boldsymbol{u}}\ \underbrace{-\dfrac{1}{\mathcal{V}_a}\,A_{aa^B}\dfrac{\alpha\,h_a\,c}{2(D+2)}\left[\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|} + 2\left(\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\cdot\boldsymbol{n}_B\right)\boldsymbol{n}_B\right]}_{\text{artificial viscosity}}, \end{cases} \tag{3.106}$$

2. Slip BC, $\boldsymbol{u}_{aa^B} = (2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B)\,\boldsymbol{n}_B$

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = -\dfrac{\rho_a}{\mathcal{V}_a}A_{aa^B}\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B + \dfrac{1}{\mathcal{V}_a}A_{aa^B}\,\dfrac{\Delta t\,\rho_a}{\rho_0}\,\dfrac{\rho_a}{2}\,\dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aa^B}\left[\rho_a\left(\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right)\left(\dfrac{\boldsymbol{n}_B}{2} - \dfrac{\boldsymbol{c}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\right) + p_a\,\boldsymbol{n}_B\right] \\ \qquad\qquad -\dfrac{\nu}{\mathcal{V}_a}A_{aa^B}\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|} - \dfrac{1}{\mathcal{V}_a}\,A_{aa^B}\dfrac{\alpha\,h_a\,c}{2(D+2)}\,\dfrac{3\,\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}, \end{cases} \tag{3.107}$$

---

[8] From the no-penetration BC, the normal velocity to the wall should show very small variations

3.4.3 Concave solid boundaries

When a particle is in the proximity of two or more adjacent concave faces (Appendix D.7-D.8), the previous approach of simply adding extra points is not feasible. We know that the resulting cell will be concave, but the Voronoi algorithm can generate only convex cells. In fact, if we still add the boundary virtual points, the resulting symmetry planes will intersect the cell as shown in Fig. 3.10, which is clearly erroneous.

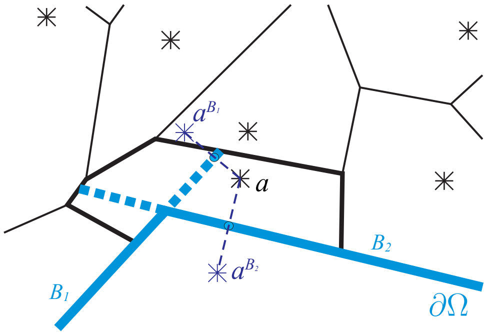


Figure 3.10: Concave cell with boundary virtual nodes

Therefore, when a particle is near two or more concave boundary faces, it does not generate any boundary virtual points. The resulting convex Voronoi cell, which extends beyond the fluid domain, is trimmed with the boundary after the tessellation is generated. The intersection algorithm used proceeds as follows:

1. Check if a particle is in the proximity of boundaries but has no boundary virtual points. We'll refer to particles that satisfy this condition as *concave boundary particles*. If not, no trimming is required and goes to the next particle.

2. Extracts the faces shared with neighbors that are also concave boundary particles, as those are the only ones that need to be trimmed.

3. Faces linked to neighbors with lower index have been trimmed already, so directly imports them.

4. Determines which cell nodes are within the fluid domain, and if a boundary node falls within the cell (the maximum particle size was limited to ensure that only one boundary node can be covered by a particle).

5. Removes cell faces with all their nodes outside the fluid domain.

6. With the remaining cell faces, determines the intersection points between cell edges and boundary faces as shown in Appendix D.9. We define edges as segments connecting two nodes in one face. In 3D and higher dimensions, cell faces and edges are different, so the intersection points between cell faces and the boundary edges need also to be calculated.

7. Based on the intersection points found, trims the existing cell faces. The possible boundary node within the cell and the intersection points are also used to create the new faces on the boundary that close the cell. Each resulting face should still have only D nodes, so more than one face can be generated on the original cell face/boundary planes.
8. Updates the trimmed faces in the tessellation data, and adds the new faces and nodes found.
9. Updates the linked faces to the trimmed cell, and starts in step 1 with the next particle.

The resulting tessellation includes degenerate cells with concave faces, but maps correctly boundaries as shown in Fig. 3.11 The next step is enforcing properly the BCs on them.

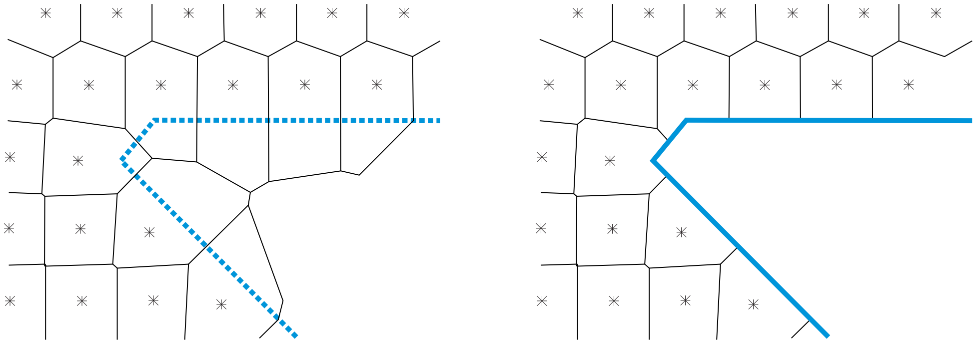

(a) Untrimmed cells (b) Trimmed cells

Figure 3.11: Voronoi diagram trimmed with concave boundary

As before, let's use a sample particle a next to a concave boundary B. As depicted in Fig. 3.12, the fluid particle within the cell can be on either side of the plane containing the concave boundary. In fact, as the particle moves, it can cross this plane without crossing or even being close to the boundary itself. This implies that at some point it could coincide with its potential boundary virtual point, which mathematically implies that $\boldsymbol{u}_{aa^B} \to \boldsymbol{0}$ and $\|\boldsymbol{r}_{aa^B}\| \to 0$. This scenario leads to a singularity in the formulation used with convex boundaries (Eqs. (3.106), (3.107)), making it numerically unstable and therefore invalid to be applied directly.

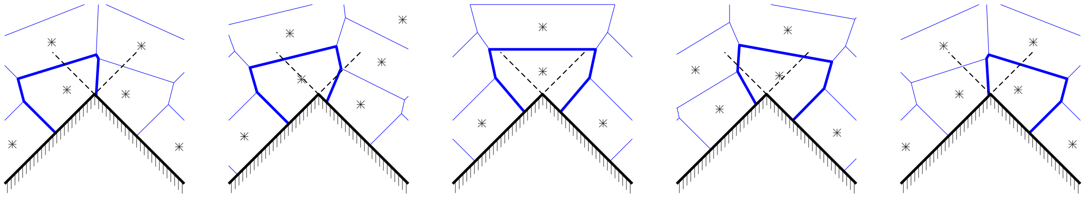

(a) (b) (c) (d) (e)

Figure 3.12: Possible scenarios of a fluid particle next to a concave boundary

Using the minimum distance to the surrounding boundaries, d_a (Appendix D.5), and the orthogonal distance to the boundary B (Appendix D.4),

$$d_{a\perp B} = (\boldsymbol{r}_a - \boldsymbol{x}_1) \cdot \boldsymbol{n}_B\,, \tag{3.108}$$

where $\boldsymbol{x}_1$ is the position vector of the closest node of the face, we can group the possible locations of the particle within the cell into three main categories, as graphically shown in Fig. 3.13.

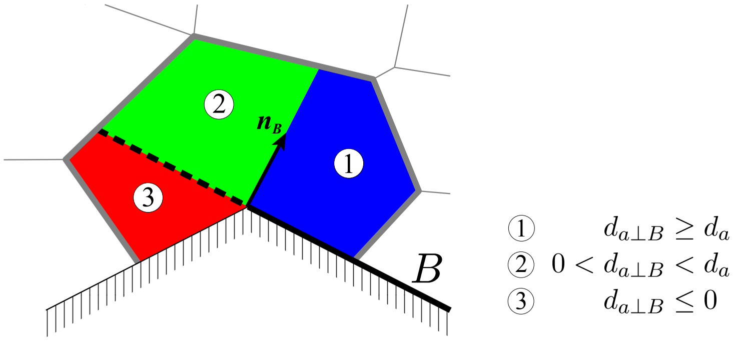


Figure 3.13: Zones within the cell into which we can categorize the position of the particle, based on d_a and $d_{a\perp B}$

Based on this subdivision, the following generalized formulation is proposed, tailored for each category:

$d_{a\perp B} \geq d_a$

Particle on the side of the boundary towards the fluid, since $\boldsymbol{n}_B$ is oriented towards the fluid. Furthermore, it falls within the orthogonal projection of the boundary so it is feasible to use the convex formulation already derived (Eqs. (3.106)-(3.107)) by simply creating a boundary virtual point at

$$\boldsymbol{r}_{a^B} = \boldsymbol{r}_a - 2\,d_{a\perp B}\,\boldsymbol{n}_B \implies \begin{cases} \boldsymbol{r}_{aa^B} = 2\,d_{a\perp B}\,\boldsymbol{n}_B\,, \\ \|\boldsymbol{r}_{aa^B}\| = 2\,d_{a\perp B}\,. \end{cases} \tag{3.109}$$

$0 < d_{a\perp B} < d_a$

Still on the fluid side, but at risk of approaching the concave boundary plane without approaching the boundary face. If we go back to the derivation of the differential operators in Appendix B, we see how

- we used the mid-point between particles as the representative point on the face, $\frac{\boldsymbol{r}_{ba}}{2}$,
- we approximated the derivative on the face as constant and evaluated it at this mid-point using the field values from the particles (Eq. (B.12)): $\nabla f(\boldsymbol{r}^*) \approx \frac{f(\boldsymbol{r}_b)-f(\boldsymbol{r}_a)}{\|\boldsymbol{r}_b-\boldsymbol{r}_a\|}\,\boldsymbol{e}_{ba} = \frac{f_{ba}}{\|\boldsymbol{r}_{ba}\|}\,\boldsymbol{e}_{ba}$.

We can revise this formulation for concave boundaries by realizing that this mid-point is not the representative point in this case. A more logical option is choosing the closest boundary node to the particle, $\boldsymbol{x}_1$, as shown in Fig. 3.14, to get a better estimate of the derivative:

$$\nabla f(\boldsymbol{r}^*) \approx \frac{f(\boldsymbol{r}_a) - f(\boldsymbol{x}_1)}{\|\boldsymbol{r}_a - \boldsymbol{x}_1\|}\,\frac{\boldsymbol{r}_a - \boldsymbol{x}_1}{\|\boldsymbol{r}_a - \boldsymbol{x}_1\|}\,. \tag{3.110}$$

Unlike with regular faces between neighbors, the field value at the face is known in this case. Following our notation convention, we define $f_{ax_1} = f(\boldsymbol{r}_a) - f(\boldsymbol{x}_1)$ and $\boldsymbol{r}_{ax_1} = \boldsymbol{r}_a - \boldsymbol{x}_1$. In

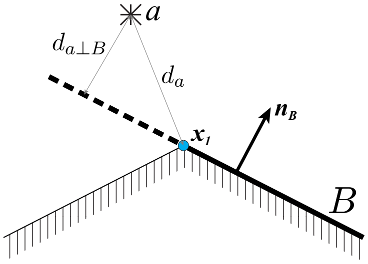


Figure 3.14: Representative point in concave faces

addition, if we reached this case, the fluid particle falls outside of the orthogonal projection of the boundary. The distance from the particle to $\boldsymbol{x}_1$ is precisely the minimum distance from the particle to the boundary, d_a. Furthermore, its projection to $\boldsymbol{n}_B$ should be precisely the orthogonal distance, $d_{a\perp B}$:

$$\|\boldsymbol{r}_{ax_1}\| = d_a\,, \tag{3.111}$$

$$\boldsymbol{r}_{ax_1}\cdot\boldsymbol{n}_B = d_{a\perp B}\,, \tag{3.112}$$

so we can write the gradient in a more compact way as

$$\nabla f(\boldsymbol{r}^*) \approx \frac{f_{ax_1}}{d_a}\frac{\boldsymbol{r}_{ax_1}}{d_a}\,. \tag{3.113}$$

Using these new expressions in Eq. (B.13), as originally done to derive the general VPH differential operators, we arrive at

$$\begin{aligned}\int_{\mathcal{S}_{ab}}\left(\frac{\boldsymbol{r}_{ba}}{2}+\boldsymbol{r}^*_\perp\right)\frac{f_{ba}}{\|\boldsymbol{r}_{ba}\|}\boldsymbol{e}_{ba}\cdot\boldsymbol{e}_{ba}\mathrm{d}\mathcal{S}^* &\rightarrow \int_{\mathcal{S}_{aB}}\left(-d_{a\perp B}\,\boldsymbol{n}_B+\boldsymbol{r}^*_\perp\right)\frac{f_{ax_1}}{d_a}\frac{\boldsymbol{r}_{ax_1}}{d_a}\cdot(-\boldsymbol{n}_B)\mathrm{d}\mathcal{S}^*\\ &= f_{ax_1}\frac{d_{a\perp B}}{d_a^2}\int_{\mathcal{S}_{aB}}\left(d_{a\perp B}\,\boldsymbol{n}_B-\boldsymbol{r}^*_\perp\right)\mathrm{d}\mathcal{S}^*\,.\end{aligned} \tag{3.114}$$

where we use the fact that $\frac{\boldsymbol{r}_{ba}}{2} = -d_{a\perp B}\,\boldsymbol{n}_B$, since it is the relative location of the orthogonal projection in the face. Comparing it with the original formulation, we observe that

$$\left(\frac{\boldsymbol{e}_{ab}}{2}-\frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right) \quad\rightarrow\quad \frac{d_{a\perp B}}{d_a^2}\left(d_{a\perp B}\,\boldsymbol{n}_B-\boldsymbol{c}_{aB}\right) \tag{3.115}$$

Note that both formulations converge as $d_{a\perp B}\rightarrow d_a$, *i.e.* the particle approaches the orthogonal projection of the boundary, since $f_{aB}\rightarrow\frac{f_{ab}}{2}$, but now does not become singular as $d_{a\perp B}\rightarrow 0$. Using in Eq. (3.61) the new approximation of the gradient at the concave boundary given in Eq. (3.113):

$$\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} = \nabla\boldsymbol{u}\cdot\boldsymbol{e}_{ab} \rightarrow \frac{\boldsymbol{u}_{ax_1}}{d_a}\frac{\boldsymbol{r}_{ax_1}}{d_a}\cdot\boldsymbol{n}_B = \frac{d_{a\perp B}}{d_a^2}\boldsymbol{u}_{ax_1}\,, \tag{3.116}$$

which is directly applied to the Laplacian and the gradient of the divergence operators in Eq. (3.62) and Eq. (3.69), respectively. Note that we have used $\boldsymbol{u}_{ax_1}$ based on the velocity of the fluid at the boundary, $\boldsymbol{u}_{x_1}$, which under the slip BC is not necessary equal to boundary velocity $\boldsymbol{u}_B$.

Finally, the pressure at the boundary in Eqs. (3.103)-(3.105) and the diffusive term from the δ-VPH correction in Eqs. (3.73)-(3.74) need to be modified. Again, using the cell face centroid for the hydrostatic correction combined with Eq. (3.115) leads to:

$$p_B = p_a - \rho_a\, \boldsymbol{g} \cdot (d_{a\perp B}\, \boldsymbol{n}_B - \boldsymbol{c}_{aB}) + d_{a\perp B} \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t} \cdot \boldsymbol{n}_B\,, \tag{3.117}$$

$$\frac{(p_b + \overline{\rho}_{ab}\, \boldsymbol{g} \cdot \boldsymbol{r}_{ab} - p_a)}{\|\boldsymbol{r}_{ab}\|}\, \boldsymbol{e}_{ab} \cdot \left(\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right) \to \rho_a \frac{d^3_{a\perp B}}{d^3_a} \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t} \cdot \boldsymbol{n}_B\,. \tag{3.118}$$

We can use these new expressions in Eq. (3.78) to obtain the final contributions for the concave boundaries to the time derivatives:

1. No-slip BC, $\boldsymbol{u}_{ax_1} = \boldsymbol{u}_a - \boldsymbol{u}_B = \boldsymbol{u}_{aB}$

$$\left\{ \begin{aligned} \left(\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} &= \overbrace{-\frac{\rho_a}{\mathcal{V}_a} A_{aB}\, \boldsymbol{u}_{ax_1} \cdot \frac{d_{a\perp B}}{d^2_a}\, (d_{a\perp B}\, \boldsymbol{n}_B - \boldsymbol{c}_{aB})}^{\rho\nabla\cdot\boldsymbol{u}} + \overbrace{\frac{1}{\mathcal{V}_a} A_{aB} \frac{\Delta t\, \rho^2_a}{\rho_0} \frac{d^3_{a\perp B}}{d^3_a} \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t} \cdot \boldsymbol{n}_B}^{\text{diffusive term, } \delta-\text{VPH}}\,, \\ \left(\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} &= \overbrace{\frac{1}{\rho_a \mathcal{V}_a} A_{aB} \left[\rho_a \left(d_{a\perp B} \left(\boldsymbol{g} - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right) \cdot \boldsymbol{n}_B - \boldsymbol{g} \cdot \boldsymbol{c}_{aB}\right) \frac{d_{a\perp B}}{d^2_a}\, (d_{a\perp B}\, \boldsymbol{n}_B - \boldsymbol{c}_{aB}) + p_a\, \boldsymbol{n}_B\right]}^{-\frac{\nabla p}{\rho}} \\ &\quad \underbrace{-\frac{\nu}{\mathcal{V}_a} A_{aB} \frac{d_{a\perp B}}{d^2_a}\, \boldsymbol{u}_{ax_1}}_{\nu\nabla^2\boldsymbol{u}} \underbrace{-\frac{1}{\mathcal{V}_a} A_{aB} \frac{\alpha\, h_a\, c}{2(D+2)} \frac{d_{a\perp B}}{d^2_a} \left[\boldsymbol{u}_{ax_1} + 2\, (\boldsymbol{u}_{ax_1} \cdot \boldsymbol{n}_B)\, \boldsymbol{n}_B\right]}_{\text{artificial viscosity}}\,. \end{aligned} \right. \tag{3.119}$$

2. Slip BC, $\boldsymbol{u}_{ax_1} = (\boldsymbol{u}_{aB} \cdot \boldsymbol{n}_B)\, \boldsymbol{n}_B$

$$\left\{ \begin{aligned} \left(\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} &= -\frac{\rho_a}{\mathcal{V}_a} A_{aB} \frac{d^2_{a\perp B}}{d^2_a}\, \boldsymbol{u}_{ax_1} \cdot \boldsymbol{n}_B + \frac{1}{\mathcal{V}_a} A_{aB} \frac{\Delta t\, \rho_a}{\rho_0} \frac{\rho_a\, d^2_{a\perp B}}{d^2_a} \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t} \cdot \frac{\boldsymbol{r}_{ax_1}}{d_a}\,, \\ \left(\frac{d\boldsymbol{u}_a}{dt}\right)^{VPH}_{B\to a} &= \frac{1}{\rho_a \mathcal{V}_a} A_{aB} \left[\rho_a \left(d_{a\perp B} \left(\boldsymbol{g} - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right) \cdot \boldsymbol{n}_B - \boldsymbol{g} \cdot \boldsymbol{c}_{aB}\right) \frac{d_{a\perp B}}{d^2_a}\, (d_{a\perp B}\, \boldsymbol{n}_B - \boldsymbol{c}_{aB}) + p_a\, \boldsymbol{n}_B\right] \\ &\quad -\frac{\nu}{\mathcal{V}_a} A_{aB} \frac{d_{a\perp B}}{d^2_a}\, \boldsymbol{u}_{ax_1} - \frac{1}{\mathcal{V}_a} A_{aB} \frac{\alpha\, h_a\, c}{2(D+2)} \frac{d_{a\perp B}}{d^2_a}\, 3\, \boldsymbol{u}_{ax_1}\,, \end{aligned} \right. \tag{3.120}$$

As a final note, we should point out that even if we are using $\boldsymbol{x}_1$ as the reference point, it does not appear explicitly in the formulation above and therefore does not need to be determined.

$d_{a\perp B} \leq 0$

In this category, the particle is located behind the fluid side of the boundary. As a result, its effect in the fluid particle behavior is indirect. Evaluating it would require some kind of field interpolation within the cell to link the particle with the fluid area next to the boundary face. Nevertheless, since the face should fall further from the particle than other faces where $d_{a\perp B} > 0$, it shouldn't have a significant impact. As a first approximation, we will neglect its effect and assume $d_{a\perp B} < 0 \equiv d_{a\perp B} = 0$, which leads to a simplified formulation:

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = 0\,, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aB}\,p_a\,\boldsymbol{n}_B\,. \end{cases} \tag{3.121}$$

both for slip and no-slip BCs.

In fact, almost all convex cells satisfy $d_{a\perp B} \geq d_a$ when all particles have the same size and a shifting algorithm keeps the fluid particle close to the cell centroid, $\boldsymbol{x}_g$. The first condition leads to neighbor particles being at a similar distance from the boundary, so their faces should be fairly orthogonal to the boundary plane (small θ in Fig. 3.15). The second condition avoids the particles drifting excessively towards to the faces, and therefore avoids the regions outside of this category, marked in green in Fig. 3.15. In addition, keeping the particle close to the cell centroid allows us to assume

$$\left.\begin{matrix} d_a \approx \Delta x \\ d_{a\perp B} \approx \Delta x \end{matrix}\right\} \implies \frac{d_{a\perp B}}{d_a^2} \approx \frac{1}{d_{a\perp B}}\,.$$

Under the first category, the convex formulation is directly used. Moreover, in the unlikely event of falling under the other possible category, $0 < d_{a\perp B} < d_a$, the conservative formulation used in this case should yield very similar results. Therefore, we could extend this generalized formulation to convex boundaries and have a unified formulation for all types of boundaries.

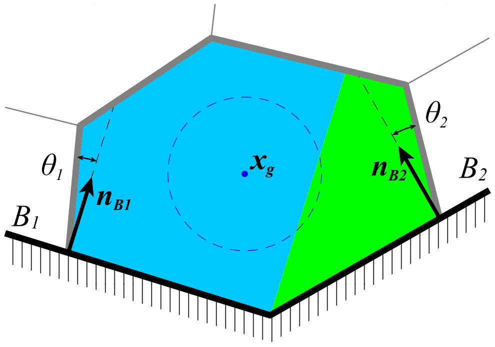


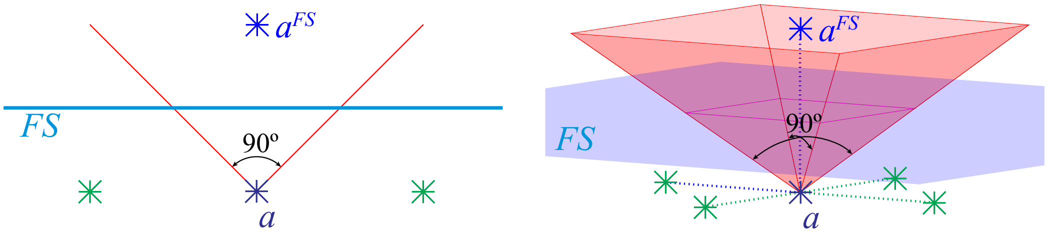
Figure 3.15: Convex cell zones for the generalized concave categories

3.4.4 FREE-SURFACE

There is not special treatment for SPH regarding the free-surface (FS). The free-surface boundary conditions (kinematic and free-stress) are a direct consequence of the Lagrangian and explicit nature of the scheme.

Regarding VPH, no previous simulations dealing with free-surfaces were found, so a specific treatment is proposed to detect whether a Voronoi element a should be bounded by a FS. The identification of free-surface shape from the particle distribution becomes increasingly difficult as we increase the dimensionality of the fluid space, allow for large deformations, surface fragmentation, air entrapment, etc. In this work, a simple approach described below is followed, leaving for future-work the exploration of more advanced methods proposed in the literature (Dilts, 2000; Haque and Dilts, 2007; Marrone et al., 2010; Marrone, 2011).

Prior to the generation of the Voronoi tessellation, the angular regions between the lines connecting the center of element a with the center of neighbor elements is explored. Under a flat free-surface, as depicted in Fig. 3.16, neighbor particles next to the FS should be located parallel to it. We can also place free-surface virtual points (FSVP) if we want to map the free-surface with Voronoi faces, as done before with the boundaries.



(a) 2D

(b) 3D

Figure 3.16: Free-surfaces and Voronoi particles

If we assign a distinct angular region for each neighbor in Fig. 3.16, without prioritizing any neighbor, we can clearly see how the portion assigned to the FS will cover a range of 90° in any angular dimension. Inspired by this fact, the algorithm proposed uses as a condition for valid directions to place FSVPs that their angle to any other particle neighbor is at least 45°,

$$\boldsymbol{e}_{a^{FS}a} \cdot \boldsymbol{e}_{ba} = \boldsymbol{e}_{aa^{FS}} \cdot \boldsymbol{e}_{ab} \leq \cos(45^\circ) \quad \forall \boldsymbol{e}_{ba} \, . \tag{3.122}$$

where $\boldsymbol{e}_{a^{FS}a} = -\boldsymbol{e}_{aa^{FS}}$ is the unit vector from particle a to a free-surface virtual point a^{FS}. The algorithm that detects FSVP directions proceeds as follows:

2D (left panel Fig. 3.17(a))
Neighbors are arranged by their angular location with respect to a. If the angle between two adjacent neighbors is greater than 90°, one or more FSVP directions (if > 180°) are assigned.

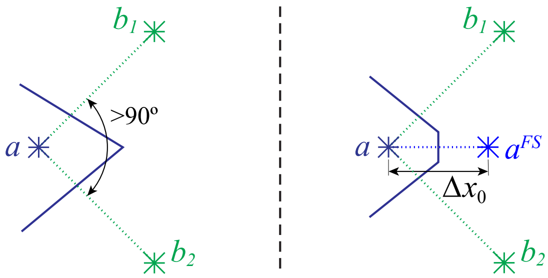
3D and higher dimensions (left panel Fig. 3.17(b))
Multiple angular dimensions exist in this case, making it difficult to keep the same procedure as in 2D. Instead, a more coarse approach is followed where we restrict the possible FSVP directions to a predefined set, each one with an assigned angular sector. The algorithm proceeds checking the sector where each neighbor falls, marking unfeasible its linked FSVP direction and the surrounding ones that fall within the ±45° threshold. The remaining feasible directions after checking all neighbors are assigned to the particle.

If the particle has no neighbors, 2^D FSVP directions aligned with the Cartesian coordinate axis are automatically assigned. Furthermore, only particles in the nearby of the free-surface identified in the previous step are checked to minimize the computational requirements.

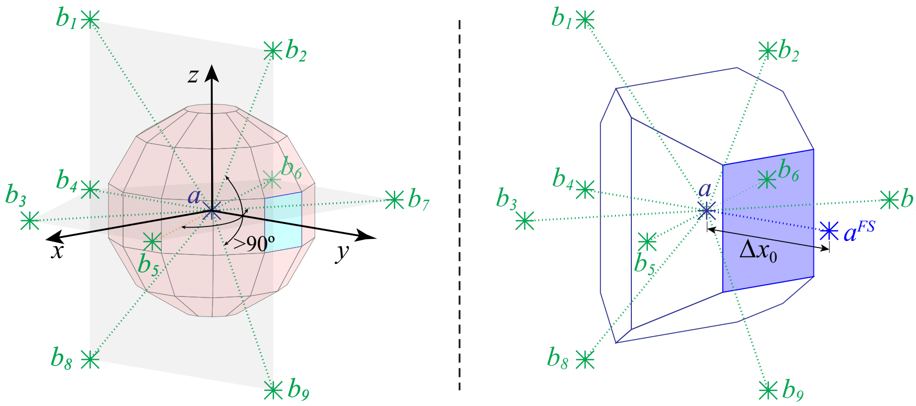
Finally, one free-surface virtual point is created for each valid direction found, and placed at a distance l defined as

$$l = \Delta x_{a,0} = V_{a,0}^{1/D} = \left(\frac{m_a}{\rho_0}\right)^{1/D} . \tag{3.123}$$

These virtual particles are considered for the Voronoi tessellation and ensure that the cells belonging to fluid particles are bounded. Defining the FSVP this way induces the creation of a face in between a and a^{FS} (right panels in Fig. 3.17), which leads to the volume of particle a being close to $\mathcal{V}_{a,0}$. Since mass is conserved, the density of such particle, ρ_a, will be close to ρ_0 and therefore the pressure at the free-surface elements will be around p_0.



(a) 2D



(b) 3D

Figure 3.17: Voronoi free surface particle detection and edge definition

The set of Voronoi faces between fluid particles and free-surface virtual particles conform the free-surface that delimit the VPH sub-domain. It is precisely in these free-surface faces where we apply the dynamic and kinematic free-surface BCs (Sec. 2.1), respectively:

$$p_{FS} = p_0 \implies p_{a^{FS}} = 2p_0 - p_a \, , \tag{3.124}$$

$$\boldsymbol{u}_{FS} = \boldsymbol{u}_a \implies \boldsymbol{u}_{aa^{FS}} = \boldsymbol{0} \, , \tag{3.125}$$

which leads to the following contribution from the free-surfaces to the time derivatives

$$\begin{cases} \left(\frac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)_{FS\to a}^{VPH} = 0\,, \\ \left(\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)_{FS\to a}^{VPH} = \frac{1}{\rho_a\,\mathcal{V}_a}\,A_{aa^{FS}}\left[p_0\,\boldsymbol{e}_{aa^{FS}} + 2\,(p_a - p_0)\,\frac{\boldsymbol{c}_{aa^{FS}}}{\|\boldsymbol{r}_{aa^{FS}}\|}\right]. \end{cases} \tag{3.126}$$

Note that $\|\boldsymbol{r}_{aa^{FS}}\| \neq 2\Delta x_{a,0}$ since a particle can interact with free-surface virtual points created by other particles.

3.4.5 VORONOI FREE-SURFACE PARTICLE CLOSE A SOLID BOUNDARY

When a free-surface SPH particle approaches a solid boundary, it is treated as a Voronoi particle, consistently with the approach discussed in Sec. 3.3.1. The same criteria to detect possible free-surface virtual point directions described in Sec. 3.4.4 is followed in the proximity of boundaries. However, in order to include the angle covered by the boundary, we also include as part of the particle neighbors:

- the linked reflected point a^B,
- the common boundary nodes if more than one face is near the particle.

However, particles already on the free surface are free to collide with or detach from a boundary. In this case no reflected point a^B is included in the free-surface criterion, leaving that region open to place a free-surface virtual point a^{FS}. While the relative distance to a^{FS} is fixed at Δx_0, the reflected point a^B gets closer as the particle a moves towards the boundary. Both virtual points are used for the Voronoi tessellation, so the algorithm that generates it will implicitly select which one (or maybe both if they are not aligned) is required to close the cell (Fig. 3.18). This approach provides a smooth transition between free surface and solid boundary conditions.

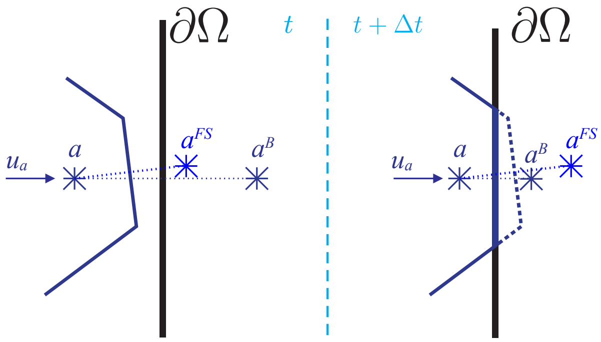


Figure 3.18: Voronoi free surface particle approaching a solid boundary

3.4.6 PERIODIC

This type of boundary condition is applied when field values at locations at $\pm\Delta\boldsymbol{r}$ are equal. Therefore, we only need to solve for the flow within a pair of planes spaced $\|\Delta\boldsymbol{r}\|$. As a result, boundary faces are placed on these planes to limit the computational domain. We can determine the periodicity direction for each face using its normal $\boldsymbol{n}_P$ such that

$$\Delta \boldsymbol{r}^P = \frac{\Delta \boldsymbol{r} \cdot \boldsymbol{n}_P}{||\Delta \boldsymbol{r} \cdot \boldsymbol{n}_P||} \Delta \boldsymbol{r} \, . \tag{3.127}$$

However, we cannot use them to bound the Voronoi cells like solid walls because physically they do not exist. Instead, we use the periodic BC to place additional layers of particles beyond the periodic faces, named a^P since each one is linked to a regular fluid particle located in the nearby of the periodic faces.

$$\boldsymbol{r}_{a^P} = \boldsymbol{r}_a + \Delta \boldsymbol{r}^P \, . \tag{3.128}$$

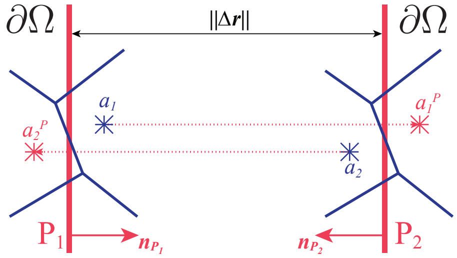


Figure 3.19: Periodic particles

Periodic particles ensure that the regular fluid particles remain bounded and are properly surrounded by neighbors. The fluid fields on them are assigned based on their linked particle:

$$\boldsymbol{u}_{a^P} = \boldsymbol{u}_a \, , \quad p_{a^P} = p_a \, . \tag{3.129}$$

3.4.7 INLETS/OUTLETS

Inlets and outlets represent the portion of the boundaries through which fluid can enter or leave the computational domain, respectively. As such, their related BCs refer either to the fluid motion through them ($\boldsymbol{u}$), or the fluid driving forces (∇p), as discussed in Sec. 2.1. In a Lagrangian scheme, particles can cross these boundaries. Moreover, since SPH and VPH rely on the interaction with neighbor particles, we need to include particles beyond inlet/outlets to ensure that the particles within the fluid domain are well surrounded (Federico et al., 2012). In the proposed coupled scheme, only Voronoi particles are in contact with the inlets/outlets to avoid mixing different particle characteristic lengths as happens in the buffer zone. This is enforced by including the distance to inlet/outlets boundaries when evaluating the criteria to determine the VPH and SPH sub-domains (Sec. 3.3.1).

Like with periodic faces, inlet/outlet boundaries do not exist physically and cannot be used to bound Voronoi cells. This closure is achieved by a layer of inlet/outlet particles, identified as b^{IO} since they act only as neighbors of regular fluid particles. In order to ensure that b^{IO} particles cover properly the fluid domain, the same algorithm that detects FSVP directions described in Sec. 3.4.4 is used on particles next to inlet/outlet faces. Every virtual point found is evaluated to check if it falls in the orthogonal projection of the inlet/outlet face (Appendix D.10), and if so is added as a inlet/outlet particle instead than a free surface virtual point as shown in Fig. 3.20.

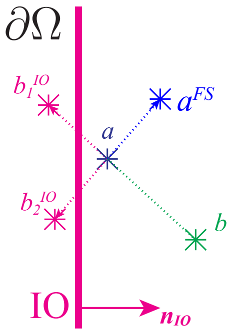


Figure 3.20: Inlet/Outlet particles generation. Particle a only detects one neighbor b and places 3 virtual points. 2 of them are beyond the I/O boundary and become I/O particles

Inlet/outlet particles are still fluid particles, so they are taken as such when acting as neighbors of particles within the fluid domain. On the other hand, the BCs are used to determine the field values on them and evaluate their movement, instead of relying on the dynamics from their neighbors. Still, these neighbors are used as a reference to extrapolate the field values using linear approximations when Neumann type BCs are applied, as done with the previous boundary types. If an inlet/outlet particle is linked to multiple regular fluid particles, the contribution from each is weighted based on the relative contact area shared such that

$$\omega_{ab^{IO}} = \frac{A_{ab^{IO}}}{A_{b^{IO}}}\,, \tag{3.130}$$

where a $A_{ab^{IO}}$ is the area of the shared face with the linked particle a, and $A_{b^{IO}} = \sum_a A_{ab^{IO}}$ is the total area shared with regular fluid particles. As a result, the following formulations are used depending on the type of inlet/outlet:

Velocity I/O

$$\boldsymbol{u}_{b^{IO}} = \boldsymbol{u}_{IO}\,, \tag{3.131}$$

$$p_{b^{IO}} = \sum_a \omega_{ab^{IO}}\, p_a\,. \tag{3.132}$$

Pressure I/O

$$\boldsymbol{u}_{b^{IO}} = \sum_a \omega_{ab^{IO}}\, \boldsymbol{u}_a\,, \tag{3.133}$$

$$p_{b^{IO}} = p_{IO}\,. \tag{3.134}$$

Free I/0

$$\boldsymbol{u}_{b^{IO}} = \sum_a \omega_{ab^{IO}}\, \boldsymbol{u}_a\,, \tag{3.135}$$

$$p_{b^{IO}} = \sum_a \omega_{ab^{IO}} p_a\,. \tag{3.136}$$

3.5 DENSITY RE-INITIALIZATION ALGORITHM

Mass is automatically conserved in this Lagrangian scheme as it is lumped in the particles. Volume is obtained from the density that is evolved using the continuity equation, with no explicit conservation requirements under the weakly compressible assumption. However, the numerical corrections and errors in the integration of the continuity equation allow the particles to "expand" without actually pushing away their neighbors accordingly. In fact, the total volume of the fluid domain that enclose the particles can differ significantly from the summation of the individual volumes of each particle. Furthermore, this incorrect expansion converts elastic into kinetic energy, as discussed later in Sec. 5, agitating the particles and leading to erroneous density, pressure, and velocity fields.

Following Colagrossi and Landrini (2003), the density field is periodically re-initialized to recover the consistency between the mass, density, and volume. Under the coupled scheme,

$$\rho_a = \begin{cases} \check{\rho}_a^{SPH} & \text{SPH zone,} \\ \check{\rho}_a^{Vor} & \text{Voronoi zone,} \\ \omega\,\check{\rho}_a^{SPH} + (1-\omega_a)\,\check{\rho}_a^{Vor} & \text{Buffer zone.} \end{cases} \tag{3.137}$$

where $\check{\rho}$ indicates the density calculated from the particle distribution instead than evolved using continuity. In this case there is no need to average weights between particles in the buffer zone since there is no requirement for a pair-wise antisymmetric formulation. Regarding the Voronoi value, its value can be directly obtained from the cell volume:

$$\check{\rho}_a^{Vor} = \frac{m_a}{\mathcal{V}_a}\,. \tag{3.138}$$

In the case of SPH, the same approach than Colagrossi and Landrini (2003) is followed

$$\check{\rho}_a^{SPH} = \sum_b m_b W_{ab}^{\mathrm{MLS}}\,. \tag{3.139}$$

A first-order accurate moving-least-squares (MLS) kernel is used in this case to guarantee its consistency at particle level, *i.e.* satisfy $\sum_b W_{ab} = 1$, and obtain a consistent value of the density. Defining the generalized position vector $\hat{\boldsymbol{r}}_{ab}$ as

$$\hat{\boldsymbol{r}}_{ab} = \left(1, \boldsymbol{r}_{ab}^1, \dots, \boldsymbol{r}_{ab}^D\right)\,, \tag{3.140}$$

we compute the MLS kernel coefficient through

$$W_{ab}^{\mathrm{MLS}} = (\boldsymbol{\beta}_a \cdot \hat{\boldsymbol{r}}_{ab})\, W_{ab}\,, \tag{3.141}$$

where W_{ab} is the kernel used for the SPH formulation as shown in Sec. 3.1.2, and $\boldsymbol{\beta}_a$ is the vector containing the specific MLS coefficients for particle a.

$$\boldsymbol{\beta}_a = \begin{pmatrix} \boldsymbol{\beta}_a^0 \\ \boldsymbol{\beta}_a^1 \\ \vdots \\ \boldsymbol{\beta}_a^1 \end{pmatrix} = \mathbb{A}_a^{-1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{3.142}$$

with

$$\mathbb{A}_a = \sum \hat{\boldsymbol{r}}_{ab}\, \hat{\boldsymbol{r}}_{ab}\, W_{ab} \tag{3.143}$$

In the end, this procedure requires summing over all the neighbors to determine the matrix $\mathbb{A}_a$, invert it to determine the MLS coefficients $\boldsymbol{\beta}_a$, and then go again through all neighbors to calculate the density.

To avoid excessive computations, the frequency at which this density re-initialization is applied, Δn_ρ, can be adjusted as an external parameter[9]. Colagrossi and Landrini (2003) show substantial improvements in the flow fields by performing the density re-initialization every 20 steps, avoiding the high-frequency oscillations found otherwise.

3.6 SHIFTING ALGORITHM

As particles move, the resulting Voronoi cells tend to get distorted. Furthermore, a Voronoi diagram can have multiple generating particle configurations, which reflects the existence of glass modes that results in a progressive drift of the particles away from the centroid of their cell. In order to balance these effects that reduce the precision of the differential operators, a shifting algorithm inspired by the Lloyd's method (Lloyd, 1982) is used in the Voronoi sub-domain, similar to Springel (2010a). When applied, the distance d_g between the particle and the cell centroid $\boldsymbol{x}_g$ is monitored, shifting the particle if exceeds certain given threshold according to Eq. (3.144).

$$\boldsymbol{r}_a \;\rightarrow\; \boldsymbol{r}_a + \begin{cases} 0 & \text{for} \quad \dfrac{d_g}{\zeta\, \Delta x_a} < 0.9, \\ (\boldsymbol{x}_g - \boldsymbol{r}_a)\, \dfrac{d_g - 0.9\zeta\, \Delta x_a}{0.2\zeta\, \Delta x_a} & \text{for } \; 0.9 \le \dfrac{d_g}{\zeta\, \Delta x_a} < 1.1, \\ \boldsymbol{x}_g - \boldsymbol{r}_a & \text{for } \; 1.1 \le \dfrac{d_g}{\zeta\, \Delta x_a}, \end{cases} \tag{3.144}$$

where ζ is the coefficient that defines the reference threshold, typically set to $\zeta = 0.25$. The position of $\boldsymbol{x}_g$ is determined after the tessellation is generated. This shift is applied at the end of the time step to ensure that the Voronoi faces remain orthogonal to $\boldsymbol{r}_{ab}$ when computing all flow variables.

---

[9] If not defined, it is assumed that $\Delta n_\rho = 0$ and no density re-initialization is applied

CHAPTER 4

IMPLEMENTATION

4.1 OVERVIEW

The code key routines are written in C++, and the main time-stepping program is implemented in Matlab. MEX functions are used to connect both languages. Section 4.2 show graphically the program architecture. The code has been tested in 2D but is designed to work also in 3D. Parallelization has not yet been undertaken. The Voronoi tessellation requires around 20% of the computation time. Apart from it, the performance of the code is similar to that of any standard serial SPH model.

To avoid double summations, particle neighbors are identified using a hierarchical tree algorithm as commonly done in SPH (Sec. 4.3). The characteristic parameters of the geometry and initial conditions are read from text files. The initial particle distribution is generated afterwards based on a given reference particle size, Δx_0, and flow field variables are assigned to each particle based on the initial conditions. To mitigate possible inconsistencies and reach a well balanced initial state,[1] an initialization ramp can be activated (Sec. 4.4). A leap-frog numerical scheme is used to iterate forward in time, described in Sec. 4.5. The simulation proceeds until any of the following stopping criteria is reached:

- Maximum time or number of iterations reached: $t > t_{\max}$, $n > n_{\max}$, where n indicates the time step index.

- Maximum number of particles allowed exceeded: $N_p > N_{\max}$, which can occur in simulations with inlets filling the fluid domain.

- Time step increment below allowed threshold: $\Delta t < \Delta t_{\min}$

- Maximum aspect ratio of Voronoi cells beyond allowed threshold: $\max\limits_{a \in VPH} \mathit{AR}_a > \mathit{AR}_{\max}$, where $\mathit{AR}_a = \frac{d_{\max a}}{d_{\min a}}$. $d_{\max a}$ and $d_{\min a}$ are the distances from the particle to the furthest and closest Voronoi cell face, respectively. In this work we keep $\mathit{AR}_{\max} = 3$.

Values of $t_{\max}$, $n_{\max}$, $N_{p\max}$, $\Delta t_{\min}$, and $\mathit{AR}_{\max}$ are defined as external simulation parameters[2]. During the simulation, flow fields can be saved with a given periodicity defined also as a simulation parameter for post-processing later. The output data is saved as an HDF5 file[3], whose C++ library is already integrated in Matlab.

---

[1] While the fluid is modeled as weakly compressible, it can be expected to define the initial conditions as if it was incompressible. In addition, going from the continuous initial fields to a set of discrete values at the particles introduce some unbalances. As a result, undesirable excessive loads can occur at the first time steps that can destabilize the simulation.

[2] If any of these parameters is not specified, its corresponding stopping criteria is not applied

[3] Hierarchical Definition File, release 1.8.12 used. `https://www.hdfgroup.org/solutions/hdf5/`

Regarding external libraries loaded for the calculations, the *Eigen* C++ library[4] is used for matrix calculations. The generation of the Voronoi diagram is performed using the *qHull* C++ library[5]. Its performance is stated in its documentation[6]: " Let n be the number of input points, v be the number of output vertices, and f_v be the maximum number of facets for a convex hull of v vertices. If both conditions hold, Qhull runs in $O(n \log v)$ in 2-D and 3-D and $O(n f_v / v)$ otherwise. The function f_v increases rapidly with dimension. It is $O(v^{\mathrm{floor}(D/2)} / \mathrm{floor}(D/2)!)$. "

---

[4] Release 3.3.2 `http://eigen.tuxfamily.org/`

[5] Release 2005.2 `http://www.qhull.org`

[6] `http://www.qhull.org/html/qh-code.htm#performance`

4.2 CODE

Figure 4.1 below summarizes the architecture of the implementation developed. Elements highlighted in green are written in C++, red are in Matlab, and yellow combine both languages.

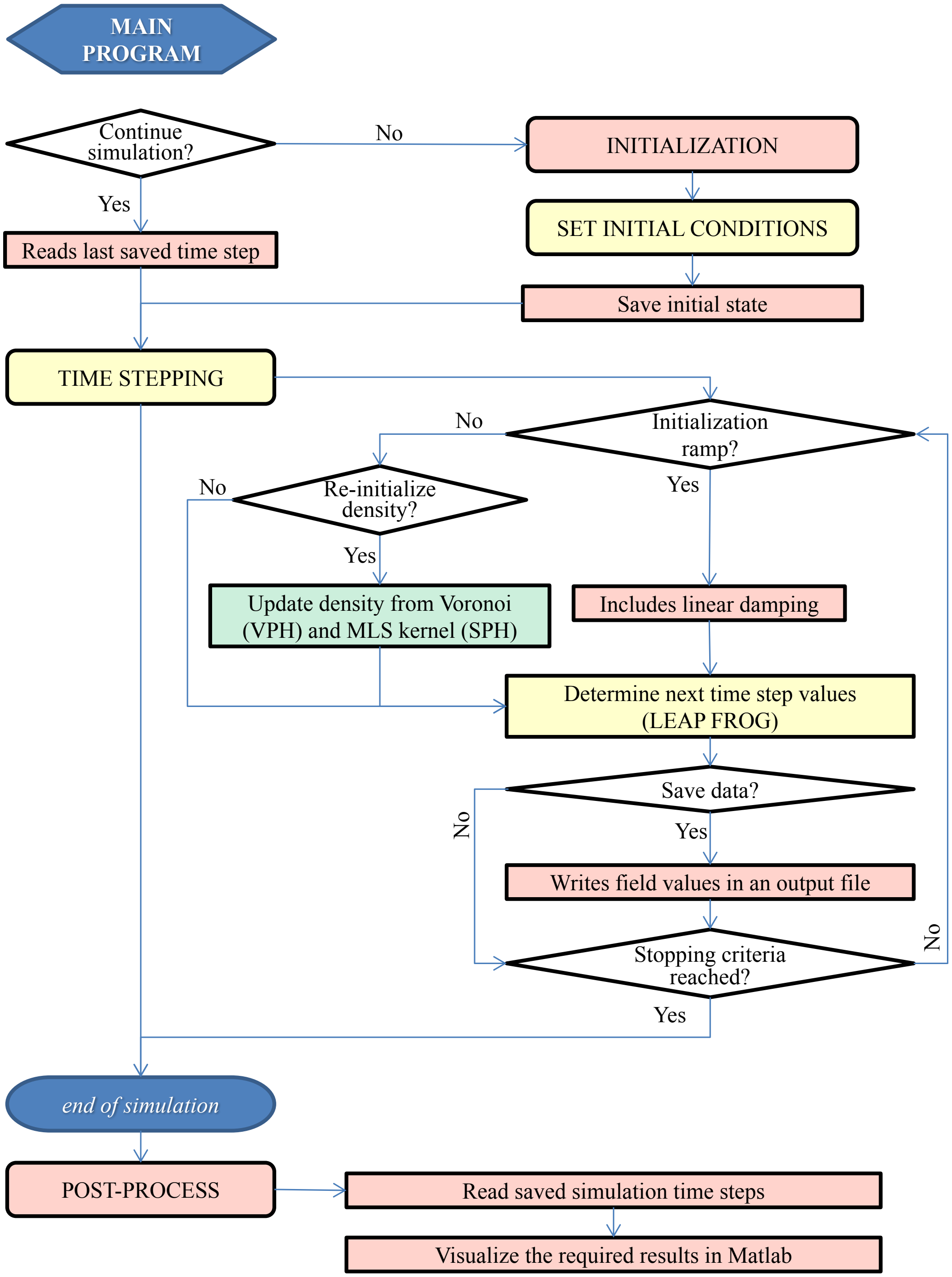


Figure 4.1: Program main structure

In addition, the block diagrams in the following figures zoom into its main modules:

- Figure 4.2: Initialization process
- Figure 4.3: Setting the initial conditions for new simulations
- Figure 4.4: Computation of particle state related variables
- Figure 4.5: Leap-frog time-stepping scheme
- Figure 4.6: Computation of time rates following the coupled scheme (Sec. 3.3)
- Figure 4.7: Contribution to the time rates from the δ-SPH scheme (Sec. 3.1.4)
- Figure 4.8: Contribution to the time rates from the δ-VPH scheme (Sec. 3.2.4)

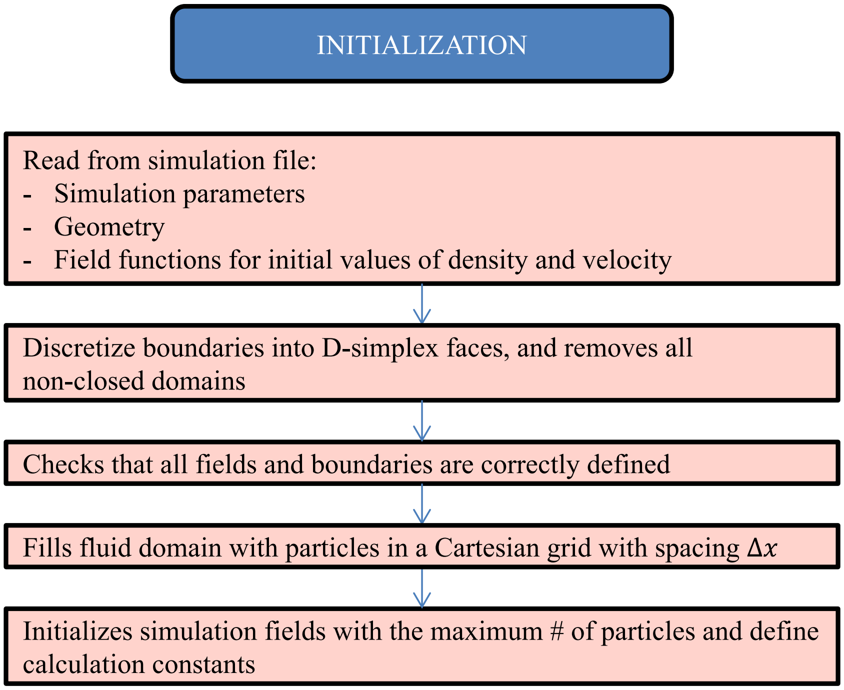


Figure 4.2: Initialization process

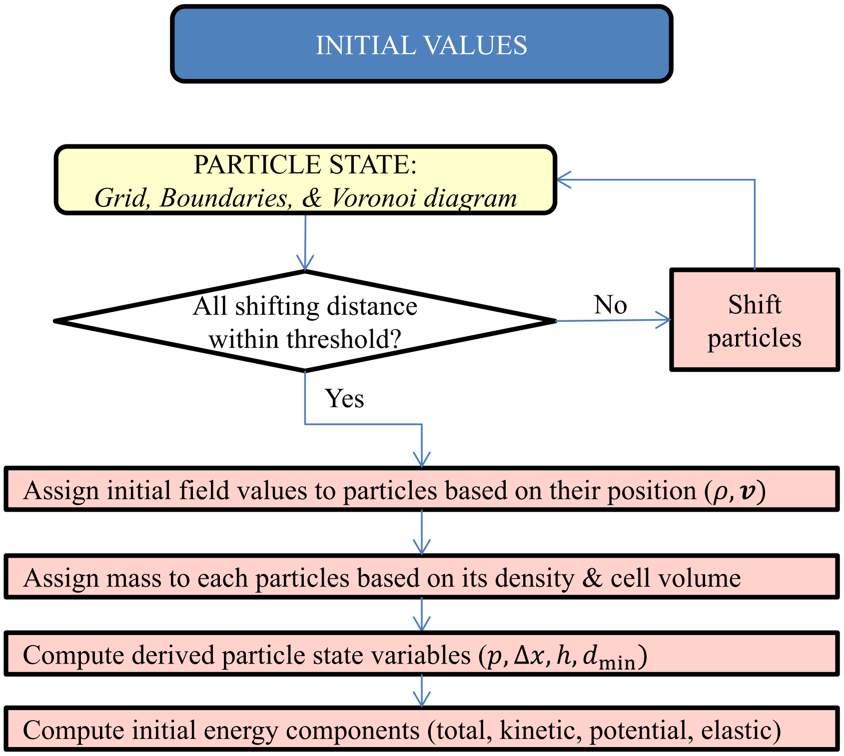


Figure 4.3: Setting initial conditions for new simulations. See Sec. 4.4 for more details

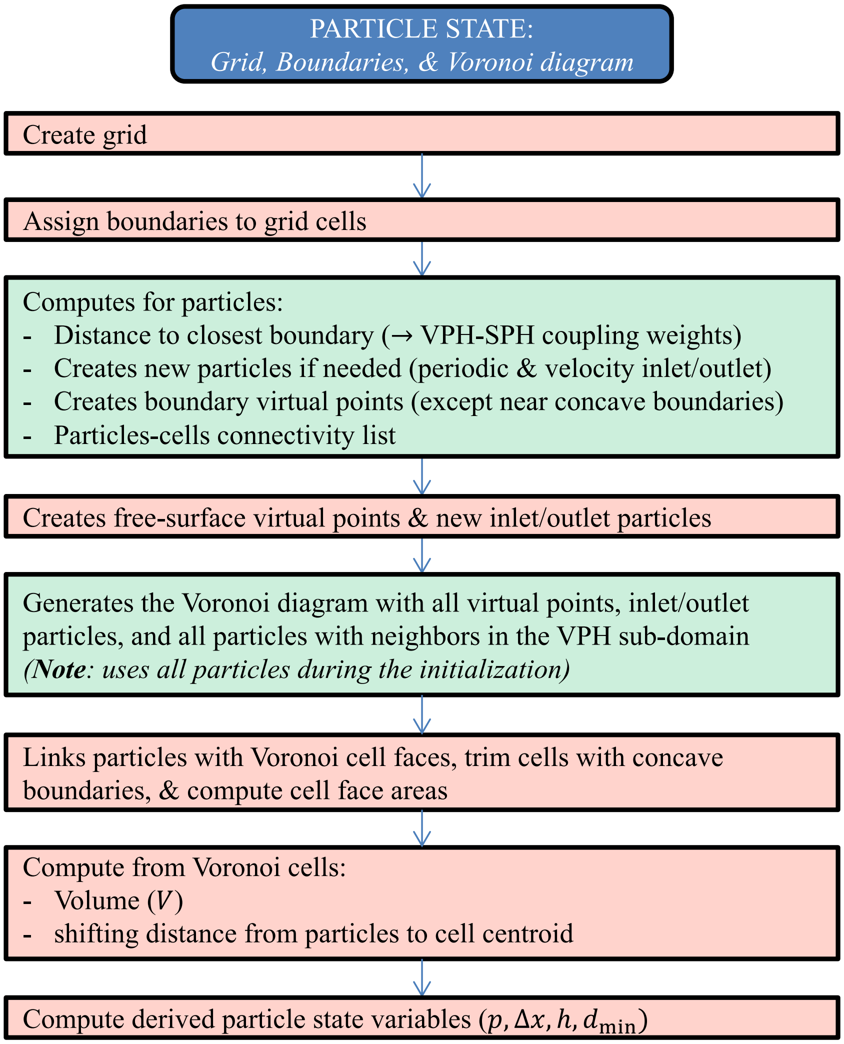


Figure 4.4: Computing particle state related variables. See Sec. 4.3 for a detailed description of how these variables are used for identifying neighbor boundaries and particles

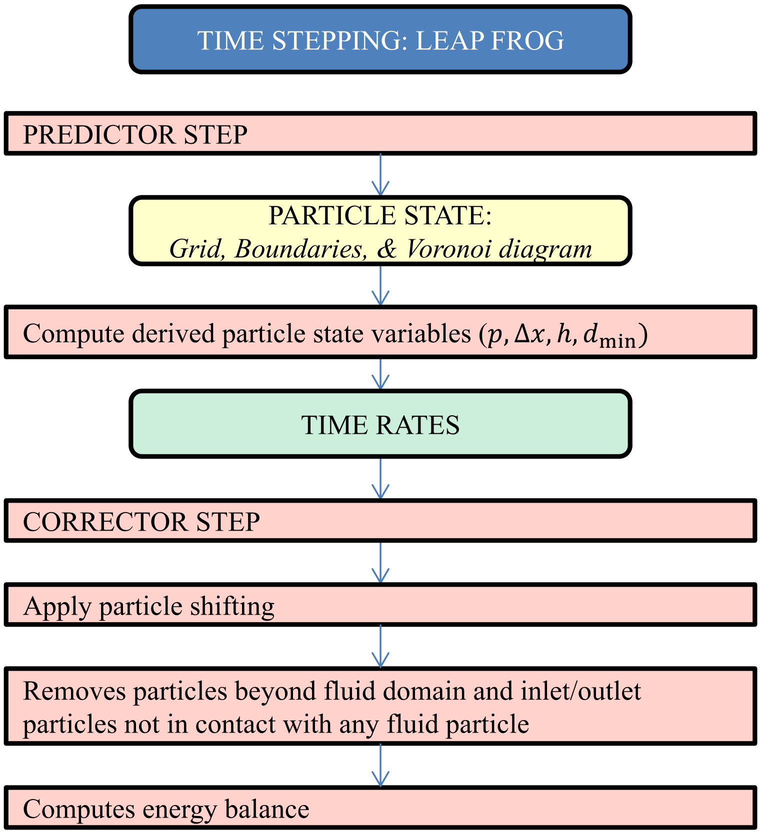


Figure 4.5: Time-stepping scheme used: Leap Frog. See Sec. 4.5 for its detailed description

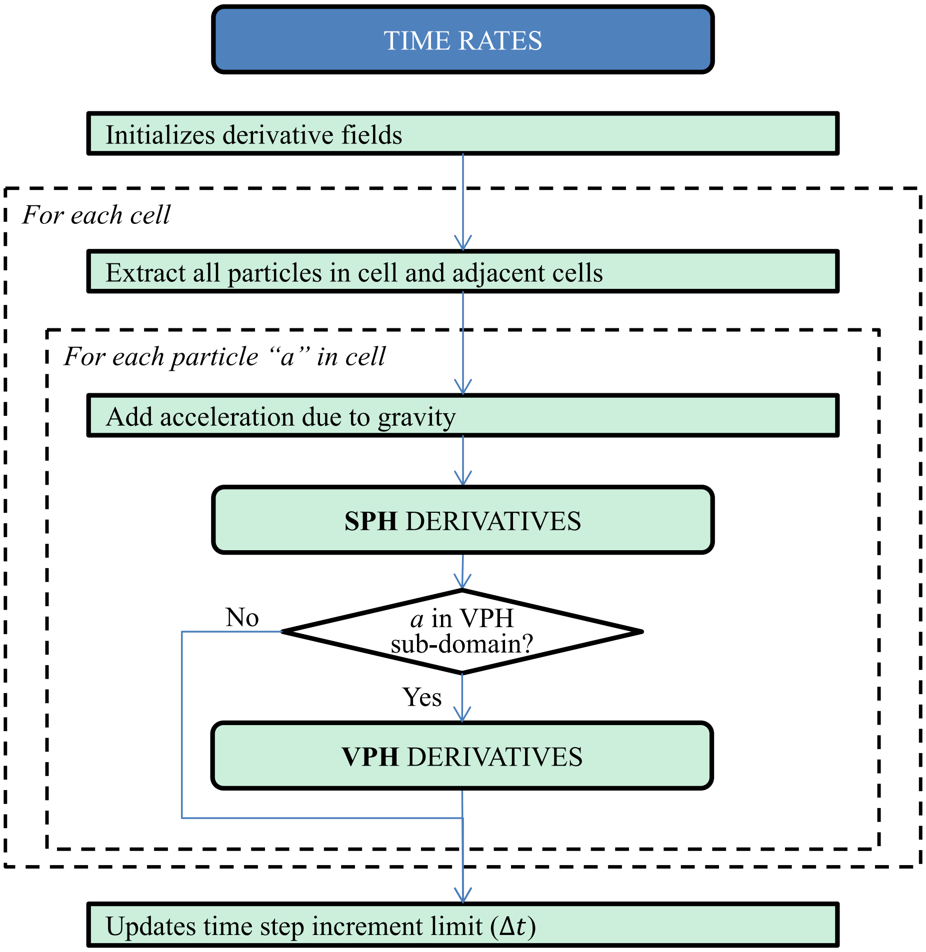


Figure 4.6: Time rates computation

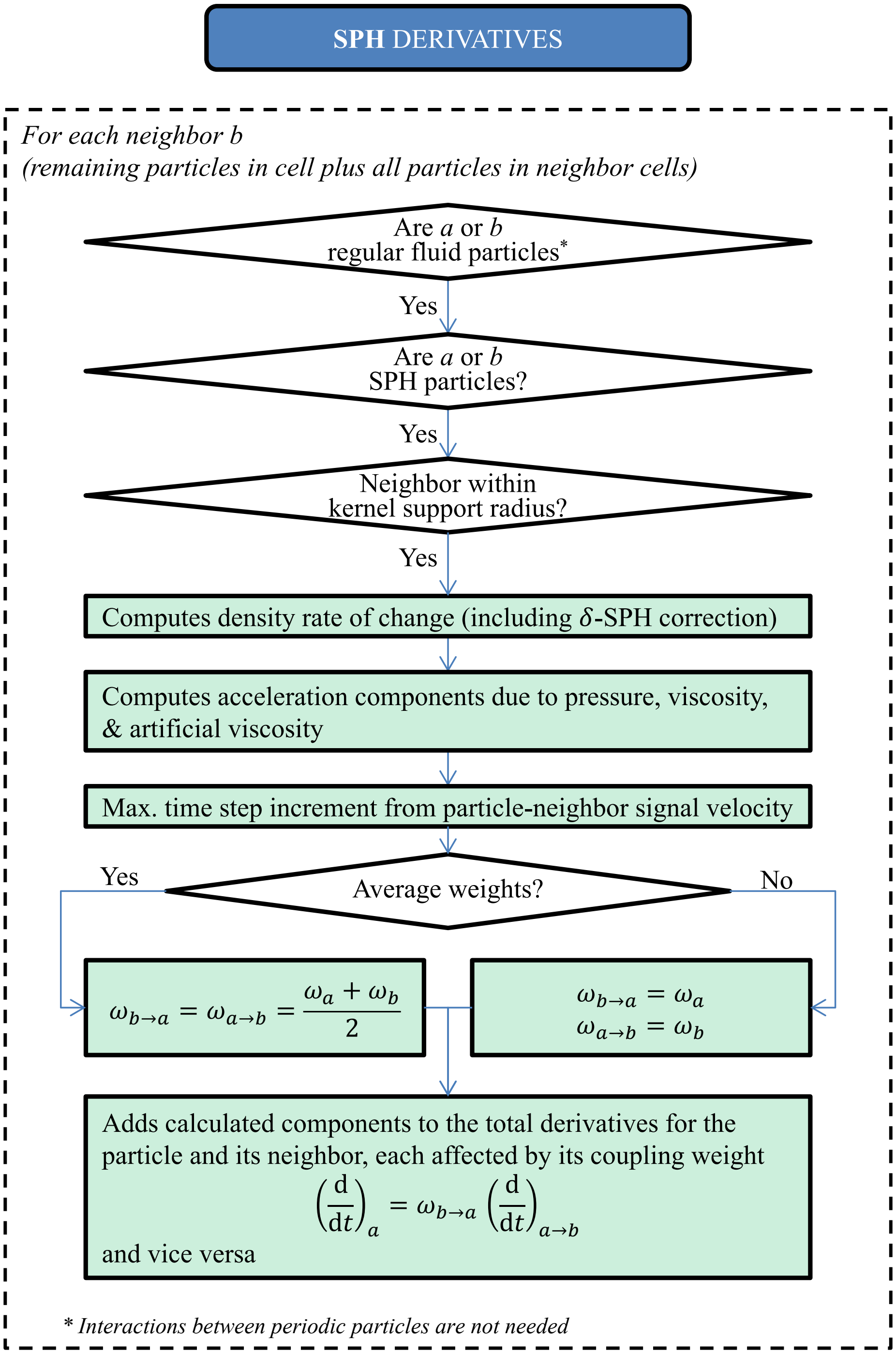


Figure 4.7: SPH time rates contribution

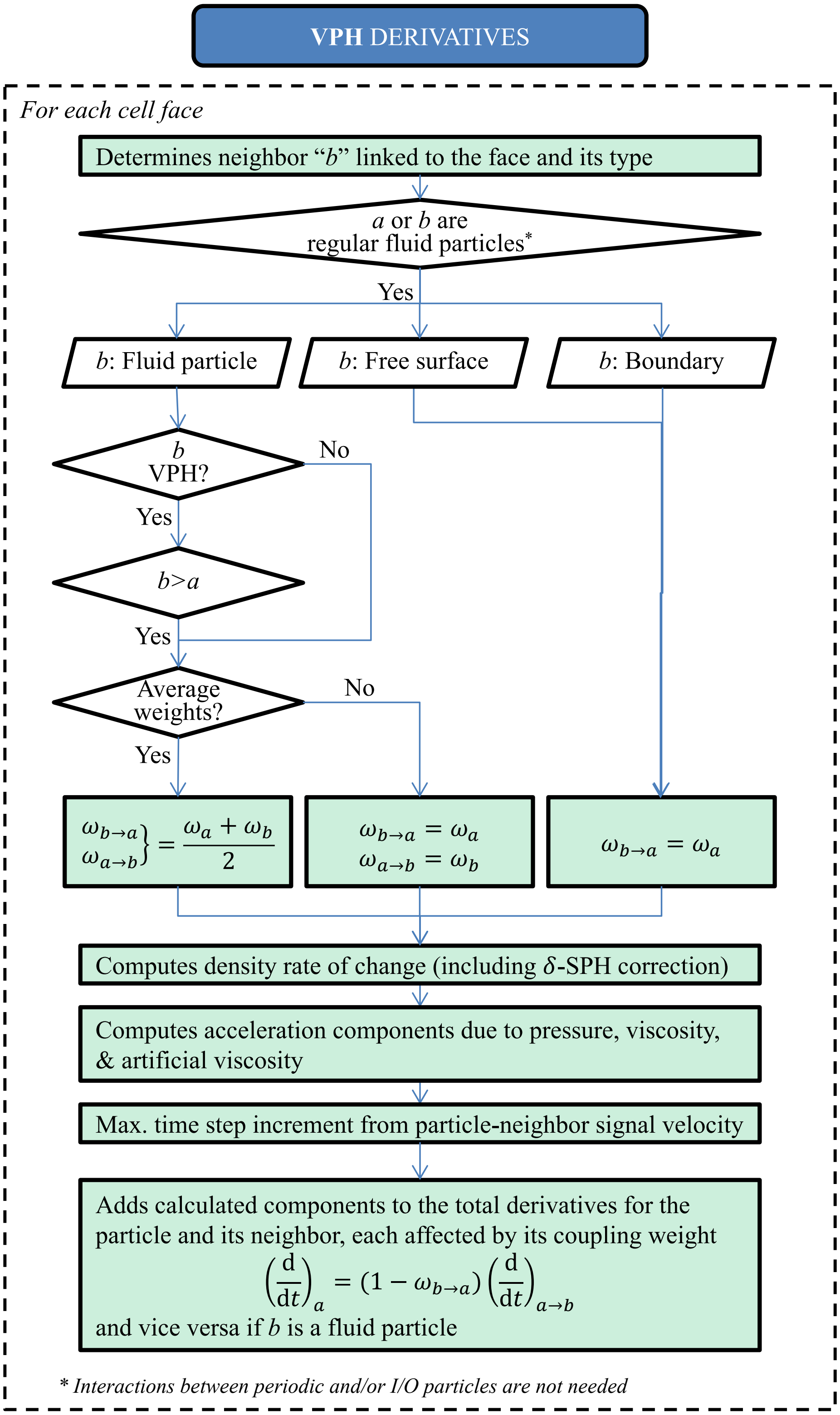


Figure 4.8: VPH time rates contribution

4.3 NEIGHBOR BOUNDARIES AND PARTICLES SEARCHING ALGORITHM

4.3.1 GRID

The range of interaction of a particle is finite in both SPH and Voronoi schemes. For this reason, an efficient way to identify the neighbors is using a hierarchical tree (Hernquist and Katz, 1989). This method relies on creating a grid that splits the fluid domain in cells such that the interaction range of the particles belonging to one grid cell will be restricted to that cell plus the neighboring ones. To achieve this, we start by creating a uniform grid at each time step whose cells must have as length $\max(\kappa h_a)$ in the case of SPH and $\max(2\, d_{\max a})$ for the VPH scheme, where

$$d_{\max a} = \max\left(\frac{\|\boldsymbol{r}_{ab}\|}{2}\right), \tag{4.1}$$

which represents the the distance to the furthest face of the Voronoi cell linked to particle a. However, we need the grid to identify the boundaries linked to each particle before creating the Voronoi tessellation, so $d_{\max a}$ is not known yet. Still, its value cannot change dramatically between time steps and is safe to assume[7] that $\max(d_{\max a})^t \le 2\max(d_{\max_a})^{t-\Delta t}$. Taking the upper bound, we determine the grid cell length as

$$l^t_{\text{grid}} = \max\left[\max_{a\in SPH}(\kappa h_a)^t,\ \max_{a\in VPH}(4\, d_{\max a})^{t-\Delta t}\right]. \tag{4.2}$$

If no value of $d^{t-\Delta t}_{\max a}$ is available, either because is the first iteration or because the particle just entered the VPH subdomain, we can approximate it using the particle size:

$$\nexists\, d^{t-\Delta t}_{\max a} \rightarrow d^{t-\Delta t}_{\max a} \approx \frac{\Delta x_a}{2}.$$

When particles with significant different sizes fall within one grid cell, it can be useful to create “child” cells within the “parent” to further remove neighbors beyond their interaction scope. This is not the case in the coupled scheme presented, where all particles have similar sizes, and therefore it is not necessary.

Once the grid is created, the next step is to assign a unique index to each grid cell. Based on the fluid domain extension and l_{grid}, we know the number of cells in each of the D Cartesian directions, $\boldsymbol{N}_c$. It is straightforward to determine the local index along each i-th direction, $\boldsymbol{k}^i_c$. Using linear indexing, we can determine a global index k_c as

$$k_c = \boldsymbol{k}^1_c + \sum_{i=2}^{D} \boldsymbol{k}^i_c\,(\boldsymbol{N}^i_c - 1). \tag{4.3}$$

---

[7] For stability reasons of the explicit scheme, we limit the time increment between time steps to ensure that the particle displacement is below a fraction of its radial size, $\Delta x_a/2$, as discussed in Sec. 4.5. Therefore, the maximum variation of $d_{\max a}$ is upper bounded by the maximum distance that two neighbor particles can move away: $\Delta \boldsymbol{r}_{ab} < (\Delta x_a + \Delta x_b)/2$. Noting that $\Delta x_a < 2\, d_{\max a}$, we conclude that

$$d^t_{\max a} \le d^{t-\Delta t}_{\max a} + \frac{\max(\|\Delta \boldsymbol{r}_{ab}\|)^{t-\Delta t}}{2} \le d^{t-\Delta t}_{\max a} + \frac{\max(\Delta x_a,\, \Delta x_b)^{t-\Delta t}}{2} \le d^{t-\Delta t}_{\max a} + \max(d_{\max a},\, d_{\max b})^{t-\Delta t}$$

$$\max d^t_{\max a} \le \max\left[d_{\max a} + \max(d_{\max a},\, d_{\max b})\right]^{t-\Delta t} = 2\max d^{t-\Delta t}_{\max a}$$

The relative local indexes to a given cell are given by all possible permutations with repetition of $\{-1, 0, 1\}$ leading to vectors of D components, except $\mathbf{0}$ that corresponds to the reference cell itself. We'll refer to this set as $\mathcal{K}_{cc}$, which contains $(3^D - 1)$ elements:

$$\mathcal{K}_{cc,2D} = \{[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]\} \,,$$
$$\mathcal{K}_{cc,3D} = \{[-1,-1,-1],[-1,-1,0],[-1,-1,1],\; \dots ,[1,1,-1],[1,1,0][1,1,1]\} \,.$$

Moreover, we can determine the global index of the j-th adjacent cell, k_{cc}^j, as

$$k_{cc}^j = k_c + \boldsymbol{\Delta k}_{cc}^j \,, \tag{4.4}$$

$$\boldsymbol{\Delta k}_{cc}^j = \boldsymbol{k}_{cc}^{1j} + \sum_{i=2}^{D} \boldsymbol{k}_{cc}^{ij} \left(\boldsymbol{N}_c^i - 1\right) \tag{4.5}$$

where $\boldsymbol{k}_{cc}^{ij}$ indicates the i-th component of the j-th element in the $\mathcal{K}_{cc}$ set, and $\boldsymbol{\Delta k}_{cc}^j$ the relative global index corresponding to this j-th element. Note that $\boldsymbol{\Delta k}_{cc}$ only depends on $\mathcal{K}_{cc}$, and therefore needs to be computed only once. This procedure to determine adjacent cells will fail with those cells on the perimeter of the grid, since not all elements in $\mathcal{K}_{cc}$ are applicable. We deal with it by extending the grid beyond the fluid domain with one additional layer of cells to ensure that all grid cells containing fluid particles are completely surrounded by adjacent cells, as shown in Fig. 4.9.

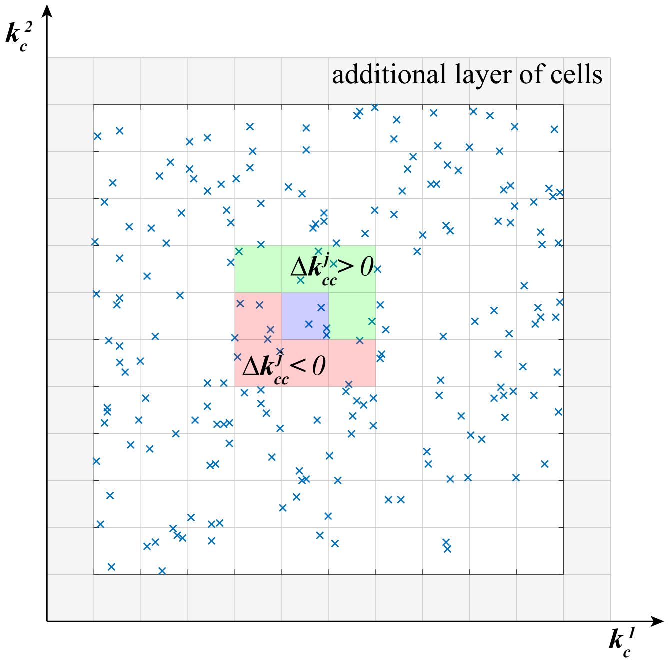


Figure 4.9: Grid (2D)

4.3.2 BOUNDARIES

Now that the grid is completely characterized, we can determine which boundary faces should be linked to each grid cell, if any. The starting point is recognizing that the domain of influence of the boundaries is precisely the VPH sub-domain, whose extension is characterized by the thickness of the Voronoi zone ($l_{\text{Vor}} > \kappa h_{\text{max}}$) and the buffer zone ($l_{\text{buffer}}$, see Sec. 3.3.3). We can express it as a multiple of the grid cell size, such that

$$N_{ccb} = \left\lceil \frac{l_{\text{Vor}} + l_{\text{buffer}}}{l_{\text{grid}}} \right\rceil \rightarrow l_{\text{VPH}} \leq N_{ccb}\, l_{\text{grid}} \,. \tag{4.6}$$

The algorithm to assign boundary faces to grid cells proceeds as follows:

1. For each boundary face, using the face nodes determine the range of grid cells covered in each dimension.
2. Enlarge it by $\pm N_{ccb}$
3. Checks for each grid cell in the extended range whether any point inside of it falls within the VPH subdomain. If so, links the cell to the boundary face.

The final outcome is a list of boundaries linked to each cell, $\mathcal{CB}$. Particles within a cell will only evaluate the distance to its associated boundaries to determine which zone do they fall in and its corresponding coupling weight value, ω. If no boundaries are linked, it belongs to the SPH zone automatically.

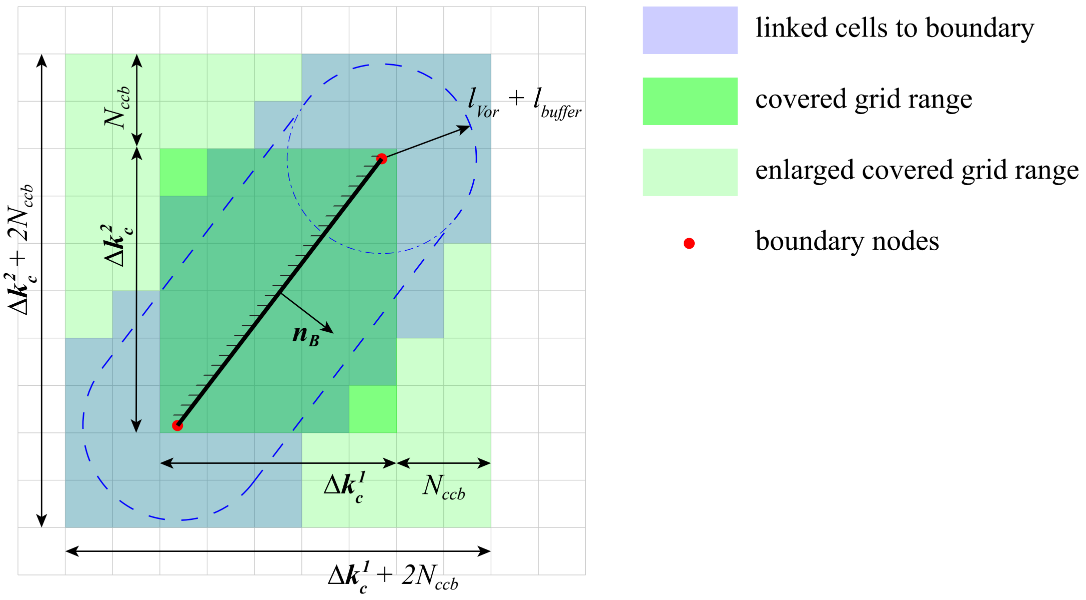


Figure 4.10: Grid cells linked to boundaries

4.3.3 CONNECTIVITY LIST

Finally, a connectivity list needs to be created to determine efficiently which particles belong to a given grid cell. This list consists of two components, graphically shown in Fig. 4.11:

- $\mathcal{CL}_P$: Particles indexes arranged sequentially by the cell that they belong to.
- $\mathcal{CL}_C$: List with the position of the first particle in $\mathcal{CL}_P$ that belongs to the cell, and the number of particles in that cell.

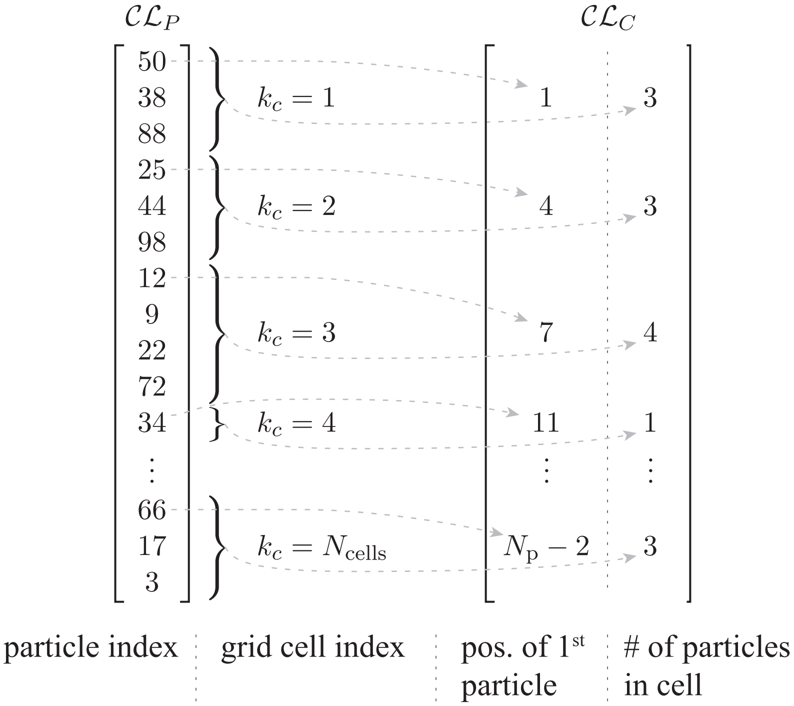


Figure 4.11: Example of a connectivity list providing the particle indexes in each cell (1-based numbering to identify the position in the array)

Loops that go through all the particles and require the interaction with neighbors, like when identifying FS particles, take advantage of this connectivity list. Instead, they are substituted by loops that go through the grid cells, limiting the interactions to the particles contained in it and its adjacent cells. Moreover, when the individual contribution from each neighbor is independent, as with the time derivatives in SPH , we can further limit the adjacent cells to those with a higher global index ($\boldsymbol{\Delta k}_{cc}^{j} > 0$ as highlighted in Fig. 4.9). Cells with a lower global index have already been analyzed, so the interaction with their particles was computed previously.

4.4 INITIALIZATION

The initialization process can either read the last saved state and continue from there, or start a new simulation based on the following input settings:

1. Geometry definition that characterizes the physical domain, $\Omega_0 := \Omega(t = 0)$, and specifies the nature of each face (solid wall, periodic, inlet/outlet) and its related variables (slip vs. no-slip, boundary velocity, etc.).
2. Initial density and velocity fields, $\rho(\mathbf{x}, t = 0)$ and $\boldsymbol{u}(\mathbf{x}, t = 0)$, gravity acceleration $\boldsymbol{g}$, physical properties of the fluid, ρ_0, p_0, ν, c, and γ (weakly compressible coefficient).
3. Reference particle size Δx_0
4. Parameters of the numerical scheme: α (artificial viscosity), whether use diffusive term in the continuity or not, whether average weights in the buffer or not, density re-initialization frequency, ζ (shifting threshold), maximum number of particles, minimum admissible time step increment, etc.
5. Frequency at which fields are plotted and saved for post processing.

Particles are initially placed on a Cartesian grid built over Ω, keeping the minimum separation with the boundaries to be approximately $\Delta x_0/2$, leading to a certain number of particles, N_p. Afterwards, a Voronoi tessellation is carried out for all particles and the shifting algorithm in Sec. 3.6 is applied repeatedly until all particles fall close to the cell centroids. Once this is achieved, the volume from the Voronoi cell for each particle is used to determine the mass of each particle, defined as $m_a = \rho(\mathbf{r}_a, 0)\,\mathcal{V}_a$, and kept constant throughout the whole simulation.

In those simulations that start from static conditions, a stabilization scheme of the particle positions is executed with the aim of finding a balanced initial condition. All viscous terms in the momentum Eq. (2.5) are replaced by a linear damping term, $-\xi\boldsymbol{u}$, to accelerate the process of reaching a static equilibrium, as proposed by Monaghan (1994):

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{g} - \frac{\nabla p}{\rho} - \xi\,\boldsymbol{u}\,, \tag{4.7}$$

where ξ controls the intensity of the damping and is typically set to $\xi\,\Delta t = 0.05$, with Δt being the time step increment. Sec. 4.5.2 shows later how is integrated in the time stepping scheme.

During the stabilization phase, the maximum velocity $u_{\max} = \max_a \|\boldsymbol{u}_a\|$ and the total kinetic energy $\mathcal{E}_K = \sum_a 0.5\,m_a\,\|\boldsymbol{u}_a\|^2$ are monitored. The stabilization evolves till $t/t_s \sim 0.1$, with t_s being the maximum time span allowed. From then on, it continues until one of the following conditions is satisfied:

- Maximum time span or maximum number of iterations allowed reached:

  $$t > t_s$$
  $$n > n_{\max s}$$

  where n indicates the time step index. Both t_s and $n_{\max s}$ are given as external parameters. Their values should be chosen depending on the nature of the flow to allow enough time for the stabilization scheme to reach an equilibrium.

- Maximum velocity satisfying

$$\frac{\max_a \|\boldsymbol{u}_a\|}{U} < 0.01$$

- Total kinetic energy satisfying

$$\frac{\mathcal{E}_K^n}{\mathcal{E}_0} < 0.001$$

- Rate of change of kinetic energy satisfying

$$\frac{\left(\mathcal{E}_K^n - \mathcal{E}_K^{n-1}\right)}{\Delta t} < 0.001\frac{\mathcal{E}_0}{t_s}$$

with U being the characteristic velocity of the problem and $\mathcal{E}_0$ a reference energy, also specified as external parameters. Once this moment is reached, the particle setup is considered in equilibrium and the simulation can be launched.

In simulations with a free surface, the background pressure p_0 of the equation of state (2.11) is set to zero, while in confined domain simulations, $p_0 := 0.05\,\rho_0\,c^2$ to make the scheme more sensitive to pressure variations while still remaining within the weakly compressible regime.

The effectiveness of this linear damping has been tested under the hydrostatic equilibrium verification case detailed in Sec. 6.1. It consists of a trapezoidal tank filled up to height H, with all particles initialized at the reference density ρ_0. Under the presence of gravity g, this case is an extension to 2D of the 1D scenario analyzed in Appendix C. Being initialized in an out-of-equilibrium situation, the fluid column is expected to show some oscillation that physically represent an exchange between kinetic and elastic energy. Figure 4.12 shows the evolution of the kinetic energy of the system of particles. It can be seen that the inclusion of the linear damping term quickly removes the acoustic-related oscillations present in the flow, bringing it rapidly to a static equilibrium.

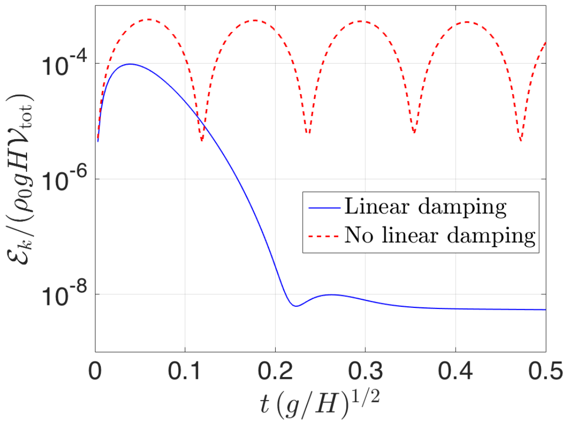


Figure 4.12: Kinetic energy evolution with and without linear damping term

4.5 TIME-STEPPING

4.5.1 TIME ITERATION SCHEME: LEAP FROG

A first-order leap frog predictor-corrector scheme is used to integrate the governing equations (Sec. 2.1) and advance in time the flow fields, following Gray (2001) and Souto-Iglesias et al. (2006). With similar computational requirements than the forward Euler method, it shows better performance because it uses information from the predicted state in the next time step. Identifying the predicted values with *, the first-order leap frog algorithm can be written for generic field A as:

$$A^{n+1} = A^n + \frac{1}{2}\left(\frac{\mathrm{d}A^*}{\mathrm{d}t} + \frac{\mathrm{d}A^n}{\mathrm{d}t}\right)\Delta t = \overbrace{A^n + \frac{\mathrm{d}A^n}{\mathrm{d}t}\Delta t}^{\text{predictor: } A^*} + \overbrace{\frac{\Delta t}{2}\left(\frac{\mathrm{d}A^*}{\mathrm{d}t} - \frac{\mathrm{d}A^n}{\mathrm{d}t}\right)}^{\text{corrector}} \tag{4.8}$$

Computationally, it is implemented as follows

1. **Predictor step.** Since velocities and accelerations are known, the positions of the particles are advanced in time with an explicit second-order scheme instead of the general first-order scheme.

$$\begin{cases} \boldsymbol{r}_a^* = \boldsymbol{r}_a^n + \boldsymbol{u}_a^n \Delta t + \dfrac{1}{2}\dfrac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\Delta t^2 \,, \\ \boldsymbol{u}_a^* = \boldsymbol{u}_a^n + \dfrac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\Delta t \,, \\ \rho_a^* = \rho_a^n + \dfrac{\mathrm{d}\rho_a^n}{\mathrm{d}t}\Delta t \,. \end{cases} \tag{4.9}$$

2. **Compute the time derivatives** following the coupled scheme formulation: $\dfrac{\mathrm{d}\boldsymbol{u}_a^*}{\mathrm{d}t}, \dfrac{\mathrm{d}\rho_a^*}{\mathrm{d}t}$.

3. **Corrector step.** Since the position was predicted with a second-order scheme, it is not corrected.

$$\begin{cases} \boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^* \,, \\ \boldsymbol{u}_a^{n+1} = \boldsymbol{u}_a^* + \dfrac{\Delta t}{2}\left(\dfrac{\mathrm{d}\boldsymbol{u}_a^*}{\mathrm{d}t} - \dfrac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\right) , \\ \rho_a^{n+1} = \rho_a^* + \dfrac{\Delta t}{2}\left(\dfrac{\mathrm{d}\rho_a^*}{\mathrm{d}t} - \dfrac{\mathrm{d}\rho_a^n}{\mathrm{d}t}\right) . \end{cases} \tag{4.10}$$

4. **Update time derivatives.** Strictly, we should recalculate them with the corrected fields, but this would duplicate the computational cost. Assuming that the corrections are small, we skip it and use the predicted values to achieve a more efficient code.

$$\frac{\mathrm{d}\boldsymbol{u}_a^{n+1}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{u}_a^*}{\mathrm{d}t} \,, \qquad \frac{\mathrm{d}\rho_a^{n+1}}{\mathrm{d}t} = \frac{\mathrm{d}\rho_a^*}{\mathrm{d}t} \,. \tag{4.11}$$

4.5.2 LINEAR DAMPING

Special mention should be made to the linear damping term followed during the stabilization phase in the initialization. Instead of incorporating it as part of the time derivatives in the δ-SPH and δ-VPH schemes, it can be added as an additional component in the time-stepping scheme. In this way, we simply need to remove the viscous terms in the momentum equation, which is simplified to

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{g} - \frac{\nabla p}{\rho} \rightarrow \begin{cases} \left(\dfrac{d\boldsymbol{u}_a}{dt}\right)^{SPH} = \boldsymbol{g}_a - \dfrac{1}{\rho_a}\displaystyle\sum_b m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right) \boldsymbol{r}_{ab}\, F_{ab}\,, \\ \left(\dfrac{d\boldsymbol{u}_a}{dt}\right)^{VPH} = \boldsymbol{g}_a + \dfrac{1}{\rho_a \mathcal{V}_a}\displaystyle\sum_{b\neq a} A_{ab} \left[(p_a + p_b)\frac{\boldsymbol{e}_{ab}}{2} + (p_a - p_b)\frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right]. \end{cases} \tag{4.12}$$

To improve its stability, we treat the linear damping term implicitly both in the predictor and corrector steps following Monaghan (1994) such that:

$$\frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t} \rightarrow \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t} - \xi \boldsymbol{u}_a^*\,,$$
$$\frac{\mathrm{d}\boldsymbol{u}_a^*}{\mathrm{d}t} \rightarrow \frac{\mathrm{d}\boldsymbol{u}_a^*}{\mathrm{d}t} - \xi \boldsymbol{u}_a^{n+1}\,.$$

Starting from the general formulation of the leap frog in Eq. (4.8), and denoting with ** the new predicted values using the linear damping term:

$$\boldsymbol{u}_a^{n+1} = \boldsymbol{u}_a^n + \frac{\Delta t}{2}\left(\frac{\mathrm{d}\boldsymbol{u}_a^{**}}{\mathrm{d}t} - \xi\boldsymbol{u}_a^{n+1} + \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t} - \xi\boldsymbol{u}_a^{**}\right) = \overbrace{\boldsymbol{u}_a^n + \Delta t\frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t} - \frac{\Delta t}{2}\xi\boldsymbol{u}_a^{**}}^{\text{predictor: }\boldsymbol{u}_a^{**}} + \overbrace{\frac{\Delta t}{2}\left(\frac{\mathrm{d}\boldsymbol{u}_a^{**}}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t} - \xi\boldsymbol{u}_a^{n+1}\right)}^{\text{corrector}}. \tag{4.13}$$

Solving each of them individually, we arrive at

$$\boldsymbol{u}_a^{**} = \frac{1}{1 + 0.5\,\xi\,\Delta t}\left(\boldsymbol{u}_a^n + \Delta t\frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\right), \tag{4.14}$$

$$\boldsymbol{u}_a^{n+1} = \frac{1}{1 + 0.5\,\xi\,\Delta t}\left[\boldsymbol{u}_a^{**} + \frac{\Delta t}{2}\left(\frac{\mathrm{d}\boldsymbol{u}_a^{**}}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\right)\right]. \tag{4.15}$$

Note that if we make $\xi \rightarrow 0$, then $\boldsymbol{u}_a^{**} \rightarrow \boldsymbol{u}_a^*$ and we recover the original scheme.

4.5.3 TIME STEP INCREMENT: CFL CONDITION

The time step increment is defined from the following criteria that affect the stability of the explicit scheme:

1. Courant condition for the propagation of sound waves, both for the first and second order terms,

$$\Delta t_1 = \min_a\left(\frac{0.5\Delta x_a}{c}\right), \tag{4.16}$$

$$\Delta t_2 = \min_a \sqrt{\Delta x_a \left(\left\|\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right\|\right)^{-1}}. \tag{4.17}$$

2. Time step increment associated with an inter-particle signal speed defined as

$$u_{ab}^{sig} = 2c - \boldsymbol{u}_{ab} \cdot \boldsymbol{e}_{ab} \,, \tag{4.18}$$

with index b going through the neighbors of particle a:

$$\Delta t_3 = \min_{ab} \left(\frac{\|\boldsymbol{r}_{ab}\|}{u_{ab}^{sig}} \right) . \tag{4.19}$$

3. Damping condition, that prevent the velocity vector to flip direction when it is larger than a given threshold, typically set as $0.01c$, such that

$$\Delta t_4 = \min_{a} \left[\|\boldsymbol{u}_a\| \left(\left\| \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} \right\| \right)^{-1} \right] \qquad \forall\, a \in \|\boldsymbol{u}_a\| > 0.01c \,. \tag{4.20}$$

The time step increment is obtained as the minimum of these, affected by a CFL factor.

$$\Delta t = CFL \, \min\left(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4\right) \,. \tag{4.21}$$

We use in this work a first-order leap frog method, so a conservative $CFL = 0.2$ was used. These limitations to the time step increment effectively bound the movement of the particle between time steps. This fact is used to estimate the grid size (Sec. 4.3.1) and the boundaries search threshold (Appendix D.8) based on the previous step inter-particle distances.

CHAPTER 5

ENERGY BALANCE

While there is literature regarding the analysis of SPH energy conservation properties (Antuono et al., 2015; Cercos-Pita et al., 2017), including the analysis of solid boundary conditions, that is, to our knowledge, not the case in VPH. However, the scheme presented in this work replicates all SPH coefficients in the VPH formulation to facilitate the coupling, enabling to extend the SPH energy balance to VPH as well.

This chapter starts, following Cercos-Pita et al. (2017), with the analysis at the continuous level in Sec. 5.1. An analytic expression for the elastic energy in a weakly compressible fluid is also derived in Sec. 5.2. Next, Sec. 5.3 extends the formulation to the discrete level to determine the different energy components under the δ-VSPH formulation. Finally, the evolution of this energy components is visualized in Sec. 5.4, using the hydrostatic tank verification case (Sec. 6.1) as an example.

5.1 CONTINUOUS LEVEL

The First Law of Thermodynamics, i.e. the conservation of energy $\mathcal{E}$, is defined for a closed system Ω as

$$\frac{\mathrm{d}\mathcal{E}_\Omega}{\mathrm{d}t} = \dot{Q} - \dot{\mathcal{W}}\,, \tag{5.1}$$

where $\dot{Q}$ is the rate of heat added to the system and $\dot{\mathcal{W}}$ the rate of work done by the system. We can distinguish between the energy of the system due to thermodynamic state, $\mathcal{E}_I$ (internal energy), and due to its motion state, $\mathcal{E}_K$ (kinetic energy), such that

$$\mathcal{E}_\Omega = \mathcal{E}_I + \mathcal{E}_K\,. \tag{5.2}$$

Regarding the rates of heat deposition and work, we can distinguish between volumetric components (body heat deposition and body forces) and the surface components (surface heat flux and boundary work). As a matter of fact, if we revert the the energy Eq. (2.3) from the Navier–Stokes formulation to its integral form, we can immediately identify each component:

$$\frac{\mathrm{d}}{\mathrm{d}t}\overbrace{\int_\Omega \rho\, e\,\mathrm{d}\mathcal{V}}^{\mathcal{E}_I} + \frac{\mathrm{d}}{\mathrm{d}t}\overbrace{\int_\Omega \rho\,\frac{1}{2}\|\boldsymbol{u}\|^2\mathrm{d}\mathcal{V}}^{\mathcal{E}_K} = \overbrace{\underbrace{\int_\Omega \boldsymbol{u}\cdot\rho\boldsymbol{f}\,\mathrm{d}\mathcal{V}}_{\substack{\text{rate of body}\\ \text{work: } \dot{\mathcal{W}}_\Omega}} + \underbrace{\int_{\partial\Omega}\boldsymbol{u}\cdot\underline{\underline{\sigma}}\cdot\mathbf{d}\mathcal{S}}_{\substack{\text{rate of surface}\\ \text{work: } \dot{\mathcal{W}}_{\partial\Omega}}}}^{-\dot{\mathcal{W}}} \overbrace{-\underbrace{\int_{\partial\Omega}\boldsymbol{q}\cdot\mathbf{d}\mathcal{S}}_{\substack{\text{surface heat}\\ \text{flux: } \dot{Q}_{\partial\Omega}}} + \underbrace{\int_\Omega \rho\dot{Q}\,\mathrm{d}\mathcal{V}}_{\substack{\text{body heat}\\ \text{flux: } \dot{Q}_\Omega}}}^{+\dot{Q}}\,, \tag{5.3}$$

Only constant gravitational effects are considered for now as body forces. Being conservative, and therefore reversible, its work is taken as a change of potential energy, $\mathcal{E}_P$. Since $\boldsymbol{u} = \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t}$,

$$-\dot{\mathcal{E}}_P = \int_\Omega \rho\, \boldsymbol{g} \cdot \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t}\, \mathrm{d}\mathcal{V} = \int_\Omega \frac{\mathrm{d}}{\mathrm{d}t} (\rho\, \boldsymbol{g} \cdot \boldsymbol{r})\, \mathrm{d}\mathcal{V}\,, \tag{5.4}$$

$$\mathcal{E}_P = -\int_\Omega \rho\, \boldsymbol{g} \cdot \boldsymbol{r}\, \mathrm{d}\mathcal{V}\,. \tag{5.5}$$

Regarding the rate of surface work, it is treated directly as a power delivered to the fluid through the boundaries, $\mathcal{P}_{\partial\Omega}$. Expanding $\mathbf{d}\boldsymbol{\mathcal{S}}|_{\partial\Omega} = \boldsymbol{n}_{\partial\Omega}\, \mathrm{d}\mathcal{S}$, we can express it as

$$\mathcal{P}_{\partial\Omega} = \int_{\partial\Omega} \boldsymbol{u} \cdot \left(\underline{\underline{\sigma}} \cdot \boldsymbol{n}_{\partial\Omega}\right) \mathrm{d}\mathcal{S} = \int_{\partial\Omega} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{F}\,, \tag{5.6}$$

where $\mathbf{d}\boldsymbol{F}$ represents the force applied on a differential boundary surface element. Moreover, we limited our scope in this research to Newtonian fluids, whose stress tensor was formulated in Eq. (2.4) as

$$\underline{\underline{\sigma}} = (-p + \lambda \nabla \cdot \boldsymbol{v})\, \mathbb{I} + 2\mu \mathbb{D}\,.$$

with $\mathbb{I}$ and $\mathbb{D} = (\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)/2$ being the identity and rate of strain tensors, respectively. Conceptually, we split $\underline{\underline{\sigma}}$ into a component associated with pressure ($-p\,\mathbb{I}$), and another with viscosity ($\underline{\underline{\tau}} = \lambda \nabla \cdot \boldsymbol{v}\, \mathbb{I} + 2\mu\, \mathbb{D}$). This subdivision leads to split $\mathcal{P}_{\partial\Omega}$ into

$$\mathcal{P}_{\partial\Omega} = \mathcal{P}^p_{\partial\Omega} + \mathcal{P}^\tau_{\partial\Omega}\,, \tag{5.7}$$

where

$$\begin{cases} \mathcal{P}^p_{\partial\Omega} := \displaystyle\int_{\partial\Omega} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{F}^p\,, & \mathbf{d}\boldsymbol{F}^p := -p\, \boldsymbol{n}_{\partial\Omega}\, \mathrm{d}\mathcal{S}\,, \\ \mathcal{P}^\tau_{\partial\Omega} := \displaystyle\int_{\partial\Omega} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{F}^\tau\,, & \mathbf{d}\boldsymbol{F}^\tau := [\lambda\, (\nabla \cdot \boldsymbol{u})_{\partial\Omega}\, \boldsymbol{n}_{\partial\Omega} + 2\mu\, \mathbb{D}_{\partial\Omega} \cdot \boldsymbol{n}_{\partial\Omega}]\, \mathrm{d}\mathcal{S}\,, \end{cases} \tag{5.8}$$

with $\mathbf{d}\boldsymbol{F}^p$ and $\mathbf{d}\boldsymbol{F}^\nu$ being the elementary pressure and viscous forces of the body surfaces acting on the fluid. Using this decomposition, we can express the force applied by the the solid boundaries as:

$$\boldsymbol{F}_{\partial\Omega_B} = \boldsymbol{F}^p_{\partial\Omega_B} + \boldsymbol{F}^\tau_{\partial\Omega_B} = \int_{\partial\Omega_B} \mathbf{d}\boldsymbol{F}^p + \int_{\partial\Omega_B} \mathbf{d}\boldsymbol{F}^\tau\,. \tag{5.9}$$

Based on the previous relations, we can summarize Eq. (5.3) as

$$\frac{\mathrm{d}\mathcal{E}_I}{\mathrm{d}t} + \frac{\mathrm{d}\mathcal{E}_K}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{E}_P}{\mathrm{d}t} + \mathcal{P}_{\partial\Omega} + \dot{Q}\,. \tag{5.10}$$

Instead of approaching $\mathcal{E}_P$ as a source of energy from external field force, it is usually considered as an internal capacity to generate that force based on the fluid position. As such, is a way of storing kinetic energy and therefore is lumped with it into a global mechanical energy, $\mathcal{E}_M$. Based on this approach, we define the total energy of the fluid system as

$$\mathcal{E}_{\text{tot}} = \mathcal{E}_I + \mathcal{E}_M = \mathcal{E}_I + \mathcal{E}_K + \mathcal{E}_P\,, \tag{5.11}$$

and express the conservation of energy equation as

$$\frac{\mathrm{d}\mathcal{E}_I}{\mathrm{d}t} + \frac{\mathrm{d}\mathcal{E}_M}{\mathrm{d}t} = \mathcal{P}_{\partial\Omega} + \dot{Q}\,. \tag{5.12}$$

However, it is not possible from the previous expressions to discern between the internal and kinetic energy evolution. In order to achieve this, we turn to the momentum Eq. (2.2). Dotting with $\boldsymbol{u}$ both sides of the equation, and integrating over the fluid domain, we arrive at

$$\int_\Omega \rho \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\cdot\boldsymbol{u}\,\mathrm{d}\mathcal{V} = \underbrace{\int_\Omega \rho\,\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{u}\cdot\boldsymbol{u})\,\mathrm{d}\mathcal{V}}_{\dot{\mathcal{E}}_K} = \underbrace{\int_\Omega \rho\,\boldsymbol{f}\cdot\boldsymbol{u}\,\mathrm{d}\mathcal{V}}_{-\dot{\mathcal{E}}_P} + \int_\Omega \left(\nabla\cdot\underline{\underline{\sigma}}\right)\cdot\boldsymbol{u}\,\mathrm{d}\mathcal{V}\,. \tag{5.13}$$

We can relate the last volume integral to the power delivered by the boundaries using the divergence theorem.

$$\begin{aligned}\int_\Omega \left(\nabla\cdot\underline{\underline{\sigma}}\right)\cdot\boldsymbol{u}\,\mathrm{d}\mathcal{V} &= \int_\Omega \nabla\cdot\left(\underline{\underline{\sigma}}\boldsymbol{u}\right)\mathrm{d}\mathcal{V} - \int_\Omega \underline{\underline{\sigma}} : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V} \\ &= \int_{\partial\Omega} \underline{\underline{\sigma}}\boldsymbol{u}\cdot\mathbf{d}\boldsymbol{\mathcal{S}} - \int_\Omega \underline{\underline{\sigma}} : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V} = \mathcal{P}_{\partial\Omega} - \int_\Omega \underline{\underline{\sigma}} : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V}\,.\end{aligned} \tag{5.14}$$

Physically, the last integral that depends on $\underline{\underline{\sigma}} : \nabla\boldsymbol{u}$ represents the energy used to deform the fluid without altering its motion state (Kundu and Cohen, 2008). Following the same decomposition into pressure and viscous terms,

$$\int_\Omega \underline{\underline{\sigma}} : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V} = \overbrace{\int_\Omega -p\,\mathbb{I} : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V}}^{\dot{\mathcal{E}}_C} + \overbrace{\int_\Omega \left(\lambda\,\nabla\cdot\boldsymbol{v}\,\mathbb{I} \,+\, 2\,\mu\,\mathbb{D}\right) : \nabla\boldsymbol{u}\,\mathrm{d}\mathcal{V}}^{\mathcal{P}_V}\,, \tag{5.15}$$

Noting that $(\mathbb{I} : \nabla\boldsymbol{u}) = \nabla\cdot\boldsymbol{u}$ and $(\mathbb{D} : \nabla\boldsymbol{u}) = \mathbb{D} : \mathbb{D}$[1],

$$\frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} = \int_\Omega -p\,\left(\nabla\cdot\boldsymbol{u}\right)\,\mathrm{d}\mathcal{V}\,, \tag{5.16}$$

$$\mathcal{P}_V = \int_\Omega \lambda\,\left(\nabla\cdot\boldsymbol{u}\right)^2 \,+\, 2\,\mu\,\mathbb{D} : \mathbb{D}\,\mathrm{d}\mathcal{V} \tag{5.17}$$

Using these expressions, we can formulate the evolution of the kinetic energy from Eq. (5.13) as

$$\frac{\mathrm{d}\mathcal{E}_K}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{E}_P}{\mathrm{d}t} + \mathcal{P}_{\partial\Omega} - \frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} - \mathcal{P}_V\,. \tag{5.18}$$

Physically, we can relate the divergence of the velocity to the density change by the continuity Eq. (2.8). As such, $\dot{\mathcal{E}}_C$ represents the portion of kinetic energy converted into elastic energy by compressing the fluid, hence its name. This is a reversible term, justifying why is treated as an

---

[1] Decomposing $\nabla\boldsymbol{u} = 0.5\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right) + 0.5\left(\nabla\boldsymbol{u} - \nabla\boldsymbol{u}^T\right)$, the anti-symmetric component cancels when dotted with $\mathbb{D}$, which is precisely the symmetric component.

energy transfer. On the other hand, $\mathcal{P}_V$ is the viscous dissipation rate. It is always positive, consistent with the Second Law of Thermodynamics, and represents the amount of kinetic energy that is permanently converted into internal energy. Lastly, we can combine Eq. (5.10) and Eq. (5.18) to isolate the evolution of the internal energy, which reads

$$\frac{\mathrm{d}\mathcal{E}_I}{\mathrm{d}t} = \frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} + \mathcal{P}_V + \dot{Q}\,. \tag{5.19}$$

Based on this last expression, we subdivide the internal energy into the elastic component already mentioned and a thermal component, $\mathcal{E}_T$. Substituted in Eq. (5.11), leads to the final decomposition of the total energy of the system as

$$\mathcal{E}_{\text{tot}} = \mathcal{E}_K + \mathcal{E}_P + \mathcal{E}_C + \mathcal{E}_T\,. \tag{5.20}$$

5.2 Elastic energy

Using the continuity Eq. (2.8), we can express the rate of change of the elastic energy in Eq. (5.16) as

$$\nabla\cdot\boldsymbol{u} = -\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} \quad\rightarrow\quad \frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} = \int_\Omega \frac{p}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t}\,\mathrm{d}\mathcal{V}\,. \tag{5.21}$$

If the fluid was incompressible, $\frac{\mathrm{d}\rho}{\mathrm{d}t} = 0$, which implies that the elastic energy remains invariant. However, under the weakly compressible regime this is not the case.

First, we substitute the equation of state for weakly compressible fluids, Eq. (2.11), in the rate of change of elastic energy, leading to

$$p = \frac{c^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] + p_0 \quad\rightarrow\quad \frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} = \int_\Omega \left(\frac{c^2}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma-1} - \frac{\rho_0}{\rho}\right] + \frac{p_0}{\rho}\right)\frac{\mathrm{d}\rho}{\mathrm{d}t}\,\mathrm{d}\mathcal{V}\,. \tag{5.22}$$

Taking as a reference the fluid at rest ($\rho(\boldsymbol{r}) = \rho_0 \rightarrow \mathcal{E}_C = 0$), we can solve the time integral to determine an expression for the elastic energy

$$\int_{\mathcal{E}_C(0)}^{\mathcal{E}_C(t)} \mathrm{d}\mathcal{E}_C^* = \int_\Omega \int_{\rho(0)}^{\rho(t)} \left[\frac{c^2}{\gamma}\left(\frac{\rho^*}{\rho_0}\right)^{\gamma-1} + \left(p_0 - \frac{c^2\rho_0}{\gamma}\right)\frac{1}{\rho^*}\right]\mathrm{d}\rho^*\,\mathrm{d}\mathcal{V}\,, \tag{5.23}$$

$$\mathcal{E}_C = \int_\Omega \frac{c^2\rho_0}{\gamma^2}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] + \left(p_0 - \frac{c^2\rho_0}{\gamma}\right)\ln\left(\frac{\rho}{\rho_0}\right)\mathrm{d}\mathcal{V}\,. \tag{5.24}$$

5.3 Discrete level

Following the particle discretization done both in SPH and VPH, we approximate the flow fields within each particle domain as constant, either smoothed or volume averaged as discussed in Secs. 3.1.1 and 3.2.1. Under this assumption, we can transform the integral expressions into summations such that, for a general integrand A,

$$\int_\Omega A\,\mathrm{d}\mathcal{V} \rightarrow \sum_a \int_{\Omega_a} A_a\,\mathrm{d}\mathcal{V} = \sum_a A_a\frac{m_a}{\rho_a}\,, \tag{5.25}$$

where a goes through all the fluid particles. Applying it to the different energy components, we arrive at their discrete formulation:

$$\mathcal{E}_K(t) = \sum_a \frac{1}{2} m_a \|\boldsymbol{u}_a(t)\|^2 \,, \tag{5.26}$$

$$\mathcal{E}_P(t) = \sum_a m_a \boldsymbol{g} \cdot \boldsymbol{r}_a(t) \,, \tag{5.27}$$

$$\mathcal{E}_C(t) = \sum_a \frac{c^2 \rho_0}{\gamma^2} \frac{m_a}{\rho_a(t)} \left[\left(\frac{\rho_a(t)}{\rho_0} \right)^{\gamma} - 1 \right] + \left(p_0 - \frac{c^2 \rho_0}{\gamma} \right) \frac{m_a}{\rho_a(t)} \ln \left(\frac{\rho_a(t)}{\rho_0} \right) . \tag{5.28}$$

In a similar way, the power delivered from the solid boundaries from Eq. (5.6) becomes

$$P_{\partial\Omega_B} = \sum_B \sum_{a \leftrightarrow B} \boldsymbol{u}_B \cdot \boldsymbol{F}_{B\to a} = \sum_B \sum_{a \leftrightarrow B} m_a \, \boldsymbol{u}_B \cdot \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{VPH}_{B\to a} \,, \tag{5.29}$$

where B goes through all the solid boundaries, $a \leftrightarrow B$ goes through the interactions between a boundary and all the particles in contact with it, and $B \to a$ indicates the action on an a-th particle due to B. Only VPH is used in the coupled scheme near the boundaries, hence the exclusive use of $\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{VPH}$. We refer to Secs. 3.4.2 and 3.4.3 for the specific formulations applied for convex and concave walls, respectively. In fact, we can follow the same approach with the rest of boundary types that enclose the fluid domain:

- Free-surfaces do not transfer any power, since we impose on them $p = 0$ (dynamic BC, no background pressure is applied when free-surfaces are involved).
- Periodic boundaries do not deliver any overall power, since the net power transferred to the fluid on one side should be equal and with opposite sign on the other side.
- Inlet/Outlets do interact with regular fluid particles, and as such contribute to $\mathcal{P}_{\partial\Omega}$. Since the contact area with the IO particles is known from the Voronoi diagram, we can follow the same approach than with solid boundaries (Sec. 3.4.7). Doing a linear approximation to determine the velocity at the face, as done for the differential operators,

  $$P_{\partial\Omega_{IO}} = \sum_{IO} \sum_{a \leftrightarrow b^{IO}} m_a \frac{\boldsymbol{u}_{b^{IO}} + \boldsymbol{u}_a}{2} \cdot \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{VPH}_{b^{IO}\to a} . \tag{5.30}$$

  Moreover, even if the mass of each particle does not change along the simulation, we must also account for particles added/removed from the fluid domain through this type of boundaries. Under the time stepping scheme, they behave as discrete variations of the total energy of the system,

  $$\Delta\mathcal{E}_{\mathrm{tot}}^{a \rightleftarrows b^{IO}} = \sum_{b^{IO}\to a} \overbrace{\mathcal{E}_{K,b^{IO}} + \mathcal{E}_{P,b^{IO}} + \mathcal{E}_{C,b^{IO}} + \mathcal{E}_{T,b^{IO}}}^{\mathcal{E}_{\mathrm{tot},b^{IO}}} - \sum_{a\to b^{IO}} \overbrace{\mathcal{E}_{K,a} + \mathcal{E}_{P,a} + \mathcal{E}_{C,a} + \mathcal{E}_{T,a}}^{\mathcal{E}_{\mathrm{tot},a}} \,, \tag{5.31}$$

  where $b^{IO} \to a$ goes through the IO particles that become fluid particles and $a \to b^{IO}$ through the fluid ones that become IO particles.

The next component is the viscous dissipation rate. Instead of solving directly Eq. (5.17), we can take an step back and notice that from Eq. (5.14) we can write

$$\int_{\Omega} \underline{\underline{\tau}} : \nabla \boldsymbol{u} \, \mathrm{d}\mathcal{V} = \int_{\partial\Omega} \underline{\underline{\tau}} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{\mathcal{S}} - \int_{\Omega} \left(\nabla \cdot \underline{\underline{\tau}} \right) \cdot \boldsymbol{u} \, \mathrm{d}\mathcal{V} \,, \tag{5.32}$$

$$\mathcal{P}_V = \int_{\partial\Omega} \boldsymbol{u} \cdot \mathbf{d}\boldsymbol{F}^{\tau} - \int_{\Omega} \rho \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{V} \cdot \boldsymbol{u} \, \mathrm{d}\mathcal{V} \,, \tag{5.33}$$

where $\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{V}$ stands for the component of the time derivative of the velocity due to the viscous effects from the momentum Eq. (2.5). Going into its discrete level, it becomes

$$\mathcal{P}_V = \sum_{\partial\Omega} \sum_{a \leftrightarrow \partial\Omega} m_a \, \boldsymbol{u}_{\partial\Omega} \cdot \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{V,\,VPH}_{\partial\Omega \to a} - \sum_{a} m_a \, \boldsymbol{u}_a \cdot \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{V}_{a} \,, \tag{5.34}$$

where we have used Eq. (5.29) to express in a general way the viscous power from the boundaries, and the time derivatives are approximated following the coupled VSPH formulation described in Sec. 3.3.[2]

However, the discrete evolution equations include additional terms that haven't been accounted for at the continuous level. Furthermore, additional corrections applied at each time step such as the density re-initialization and the particles shifting alter the particle state, and therefore should be included in the discrete energy balance. Keeping in mind that the goal of all these corrections is to improve the stability of the scheme, physically they should act overall as a dissipative coefficient that tends to reduce the total energy. Therefore, we gather all of them into a new discrete power term, $\mathcal{P}_{\mathrm{num}}$, which transfers energy into a new component, $\mathcal{E}_{\mathrm{num}}$. As a result, the conservation of total energy from Eq. (5.12) will read at the discrete level

$$\frac{\mathrm{d}\mathcal{E}_{\mathrm{tot}}}{\mathrm{d}t} = \frac{\mathrm{d}\mathcal{E}_{I}}{\mathrm{d}t} + \frac{\mathrm{d}\mathcal{E}_{M}}{\mathrm{d}t} = \mathcal{P}_{\partial\Omega} + \dot{Q} - \mathcal{P}_{num} \,, \tag{5.35}$$

$$\frac{\mathrm{d}\mathcal{E}_{\mathrm{num}}}{\mathrm{d}t} = \mathcal{P}_{\mathrm{num}} \,. \tag{5.36}$$

Under the assumption of a barotropic fluid, the pressure field is independent of the temperature field. Mathematically this implies that $\dot{Q}$ is uncoupled with $\dot{\mathcal{E}}_C$ and has no effect in the mechanical energy. For simplicity, we haven't included thermal effects in the coupled scheme yet, assuming for the simulations done that $\dot{Q} = 0$. As a result, we only solve for the mechanical energy evolution, which entitle us to restrict the numerical dissipation to the mechanical energy. Therefore, Eq. (5.18) reads at the discrete level

$$\frac{\mathrm{d}\mathcal{E}_{M}}{\mathrm{d}t} = \frac{\mathrm{d}\mathcal{E}_{K}}{\mathrm{d}t} + \frac{\mathrm{d}\mathcal{E}_{P}}{\mathrm{d}t} = \mathcal{P}_{\partial\Omega} - \frac{\mathrm{d}\mathcal{E}_{C}}{\mathrm{d}t} - \mathcal{P}_{V} - \mathcal{P}_{\mathrm{num}} \,. \tag{5.37}$$

In order to determine an expression for $\mathcal{P}_{\mathrm{num}}$, we can subdivide it into multiple components, each linked to the specific correction/approximation involved:

---

[2] In the present work we have assumed that the stress components related to the λ coefficient are neglected (see Sec. 2.1, Marrone et al. (2013), Colagrossi et al. (2013)), and therefore are not present in the computation of the time derivatives.

- $\mathcal{P}^{\alpha}_{\text{num}}$: artificial viscosity
- $\mathcal{P}^{\delta}_{\text{num}}$: diffusive coefficient (δ-correction)
- $\mathcal{P}^{\rho}_{\text{num}}$: density re-initialization
- $\mathcal{P}^{\zeta}_{\text{num}}$: particle shifting
- $\mathcal{P}^{VSPH}_{\text{num}}$: approximations for the differential operators in SPH and VPH
- $\mathcal{P}^{\Delta t}_{\text{num}}$: approximations for the time-stepping scheme used to move forward in time

The artificial viscosity appears explicitly in the momentum equation as a substitute for the fluid viscosity. Integrating it directly from the discrete momentum equation, we can determine its contribution as

$$\mathcal{P}^{\alpha}_{\text{num}} = \sum_a m_a \, \boldsymbol{u}_a \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \right)^{AV}_a \tag{5.38}$$

The derivation for the rest of the terms, however, is not trivial. Instead, since they are the last components to be determined, we lump them into a single coefficient, $\tilde{\mathcal{P}}_{\text{num}}$ and focus on its net contribution to the numerical dissipated energy.

$$\tilde{\mathcal{P}}_{\text{num}} = \mathcal{P}_{\text{num}} - \mathcal{P}^{\alpha}_{\text{num}} \, . \tag{5.39}$$

To do so, we simply compare the kinetic energy after updating the fields for a new time step with its time-evolved value without including $\tilde{\mathcal{P}}_{\text{num}}$. Starting with Eq. (5.13), we can approximate

$$\int_{\Omega} \rho \, \boldsymbol{u} \cdot \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} \, \mathrm{d}\mathcal{V} \rightarrow \sum_a m_a \, \boldsymbol{u}_a \cdot \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} \, . \tag{5.40}$$

Only the artificial viscosity numerical component is accounted for in the velocity time derivative in the coupled scheme. Therefore, from Eq. (5.18), we define the kinetic energy time derivative at the discrete level with no further numerical corrections as:

$$\frac{\mathrm{d}\mathcal{E}^{\star}_K}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{E}_P}{\mathrm{d}t} + \mathcal{P}_{\partial\Omega} - \frac{\mathrm{d}\mathcal{E}_C}{\mathrm{d}t} - \mathcal{P}_V - \mathcal{P}^{\alpha}_{num} = \sum_a m_a \, \boldsymbol{u}_a \cdot \frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} \, . \tag{5.41}$$

Following the same first-order leap frog time-stepping algorithm described in Sec. 4.5, we can evolve in time the value of the kinetic value as well

$$\mathcal{E}^{\star\,*}_K = \mathcal{E}^{n}_K + \frac{\mathrm{d}\mathcal{E}^{\star\,n}_K}{\mathrm{d}t} \Delta t \quad \rightarrow \quad \mathcal{E}^{\star\,n+1}_K = \mathcal{E}^{\star\,*}_K + \frac{\Delta t}{2} \left(\frac{\mathrm{d}\mathcal{E}^{\star\,*}_K}{\mathrm{d}t} - \frac{\mathrm{d}\mathcal{E}^{\star\,n}_K}{\mathrm{d}t} \right) + \Delta\mathcal{E}^{a \rightleftarrows b^{IO}}_K \, , \tag{5.42}$$

where n is the time step index. However, we can actually determine the kinetic energy based on the new particle state from Eq. (5.26). The difference must be due to the unaccounted numerical corrections, which lead us to conclude

$$\Delta\tilde{\mathcal{E}}_{\text{num}} = \tilde{\mathcal{E}}^{n+1}_{\text{num}} - \tilde{\mathcal{E}}^{n}_{\text{num}} = \sum_a \frac{1}{2} m_a \left\| \boldsymbol{u}^{n+1}_a \right\|^2 - \mathcal{E}^{\star\,n+1}_K \approx \tilde{\mathcal{P}}^{n}_{\text{num}} \, \Delta t \, , \tag{5.43}$$

taking as a reference the initial conditions so $\tilde{\mathcal{E}}_{\mathrm{num}}^{n=0} = 0$. Similarly, we can use the time-stepping algorithm to determine the values of the artificial and physical viscous dissipation,

$$\mathcal{E}_T^{*} = \mathcal{E}_T^{n} + \mathcal{P}_V^{n}\,\Delta t \quad \rightarrow \quad \mathcal{E}_T^{n+1} = \mathcal{E}_T^{*} + \frac{\Delta t}{2}\left(\mathcal{P}_V^{*} - \mathcal{P}_V^{n}\right) + \Delta\mathcal{E}_T^{a \rightleftarrows b^{IO}}\,, \qquad \mathcal{E}_T^{n=0} = 0\,, \tag{5.44}$$

$$\mathcal{E}_{\mathrm{num}}^{\alpha\,*} = \mathcal{E}_{\mathrm{num}}^{\alpha\,n} + \mathcal{P}_{\mathrm{num}}^{\alpha\,n}\,\Delta t \quad \rightarrow \quad \mathcal{E}_{\mathrm{num}}^{\alpha\,n+1} = \mathcal{E}_{\mathrm{num}}^{\alpha\,*} + \frac{\Delta t}{2}\left(\mathcal{P}_{\mathrm{num}}^{\alpha\,*} - \mathcal{P}_{\mathrm{num}}^{\alpha\,n}\right)\,, \qquad \mathcal{E}_{\mathrm{num}}^{\alpha\,n=0} = 0\,. \tag{5.45}$$

Note that the numerical dissipated energy is not linked to the fluid, so no variation between time steps is due to inlet/outlet particles. Also, since the coupled scheme does not account for thermal effects, it doesn't include any thermal energy associated with the particles. This simplification leads to $\Delta\mathcal{E}_T^{a \rightleftarrows b^{IO}} = 0$, which combined with $\dot{Q} = 0$ implies that $\mathcal{E}_T$ simply accumulates the overall energy dissipated due to the fluid viscosity along the simulation.

5.4 ENERGY BALANCE DURING SIMULATIONS

The same hydrostatic equilibrium configuration used in Sec. 4.4 is used now as an example to evaluate the energy components during a simulation. In fact, only the case with no linear damping term is analyzed here, since this term has not been accounted for in the energy balance.

Figure 5.1 shows the evolution of the energy components variations, defined as the difference with their corresponding initial value and made non-dimensional with the initial total energy,

$$\widehat{\Delta\mathcal{E}} = \frac{\mathcal{E} - \mathcal{E}^{0}}{\mathcal{E}_{\mathrm{tot}}^{0}}\,. \tag{5.46}$$

Time is made non-dimensional with $\sqrt{g/H}$, where g is the magnitude of the gravity acceleration.

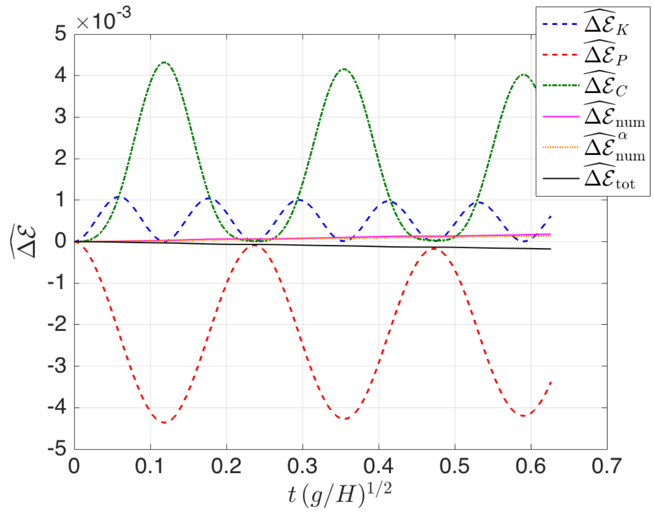


Figure 5.1: Evolution of energy components in the hydrostatic equilibrium simulation with a trapezoidal configuration (Sec. 6.1) and no linear damping

As described in Sec. 6.1, this verification case consists on a trapezoidal tank, filled to a certain height H, and initialized with the fluid at rest and $\rho(\boldsymbol{r}, t = 0) = \rho_0$. However, this density field does not correspond to the expected one from the hydrostatic pressure for a weakly compressible fluid and therefore leads to oscillations of the water column. We can clearly appreciate the exchange between kinetic, potential, and elastic energy components, with an overall decay of the total energy due to the damping terms.

The fluid is treated as inviscid ($\mathcal{P}_V = 0 \rightarrow \mathcal{E}_T = 0$), so all damping comes from the numerical dissipation ($\mathcal{P}^{\alpha}_{\text{num}}$ and $\mathcal{P}^{\sim}_{\text{num}}$). Since the fluid is initialized at rest and with the reference density ρ_0, the only non-zero initial energy component is the potential energy in this case. Therefore $\mathcal{E}^0_{\text{tot}} = \rho_0\, g\, \frac{H}{2}\, \mathcal{V}^0_{\text{tot}}$, where $\mathcal{V}^0_{\text{tot}}$ is the total volume covered initially by the fluid. Furthermore, the boundaries remain static, which implies that $\mathcal{P}_{\partial\Omega} = 0$.

CHAPTER 6

VERIFICATION

The coupled scheme presented in the previous chapters is tested with some verification cases in the following sections. They cover pressure and shear driven flows, with and without free-surfaces.

6.1 HYDROSTATIC EQUILIBRIUM: CONVEX BOUNDARIES

This test case evaluates the ability of the presented numerical scheme to reach equilibrium from an unbalanced initial condition. The same configuration than the one studied by Colagrossi et al. (2012) is analyzed here, consisting on a trapezoidal tank filled with fluid particles to a certain height and all initialized with atmospheric pressure. Under the presence of gravity, they should reach hydrostatic equilibrium if sufficient damping exists to attenuate the oscillations due to the elasticity of the fluid. Table 6.1 summarizes the simulation parameters, and $\rho(\boldsymbol{r}, t = 0) = \rho_0$.

Table 6.1: Simulation parameters: Hydrostatic equilibrium (convex boundaries)

General:	2D,	# of particles: 3125,	$\Delta x_0 = 0.02$ m,	$\boldsymbol{g} = [0, -9.81]$ m/s^2			
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	0.20 s	p_0	0 Pa	Diffusive δ-term?	✓	Average weights?	✗
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	0.02	Initialization ramp?	✓
$\Delta t_{\min}$	10^{-6} s	ν	—	ζ	0.25	t_s	0.20 s
$N_{p\,\max}$	3200	c	50 m/s	Δn_ρ	—	$n_{\max\,s}$	—
$\text{Æ}_{\max}$	3	γ	1	CFL	0.20	U	—
						$\mathcal{E}_0$	—

Figure 6.1 shows the result obtained, with no noticeable difference in the pressure field between the Voronoi and SPH sub-domains, indicating that the coupling is working adequately.

The effectiveness of the initialization linear damping is visualized through the evolution of the kinetic energy, as discussed previously in Sec. 4.4 and displayed again on the left panel of Fig. 6.2. On its right panel, the pressure is plotted as a function of the vertical position for all the particles in the fluid domain, showing reasonable agreement with the hydrostatic incompressible exact solution. One possible improvement to reduce the initial oscillation observed in Fig. 6.2 is to use the actual expected density from the hydrostatic field, ρ_{hydr}, as shown in Fig. 6.3. This hydrostatic density can be obtained from the Eq. of state (2.11) for a weakly compressible fluid as

$$\rho = \rho_0 \left[\frac{\gamma}{c^2 \rho_0} (p - p_0) + 1 \right]^{\frac{1}{\gamma}} \rightarrow \rho_{\text{hydr}}(\boldsymbol{r}) = \rho_0 \left[\frac{\gamma}{c^2 \rho_0} [\boldsymbol{g} \cdot (\boldsymbol{r} - \boldsymbol{r}_0) - p_0] + 1 \right]^{\frac{1}{\gamma}} \tag{6.1}$$

where $\boldsymbol{r}_0$ represents a location on the fluid where $p = p_0$, which in this case is $(0, H)$.

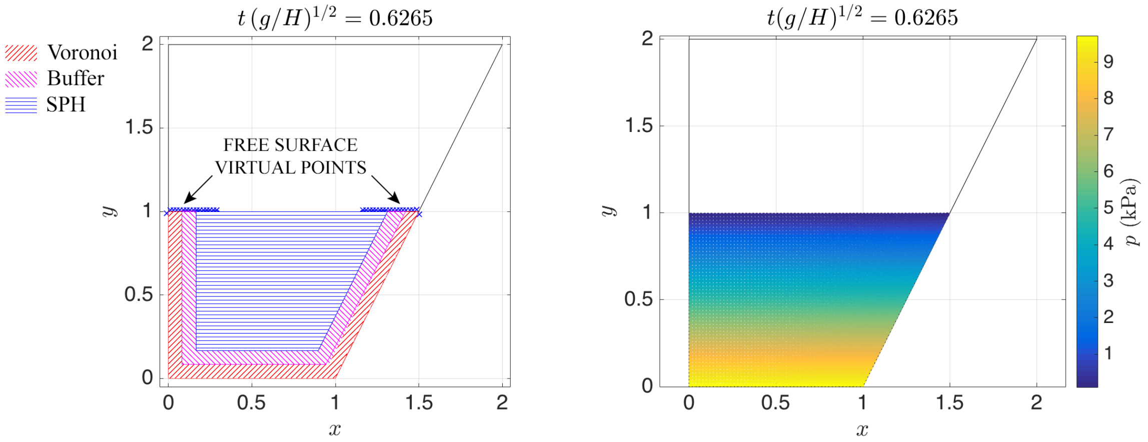


Figure 6.1: Left: Zones in a hydrostatic equilibrium simulation. Right: pressure field at same time.

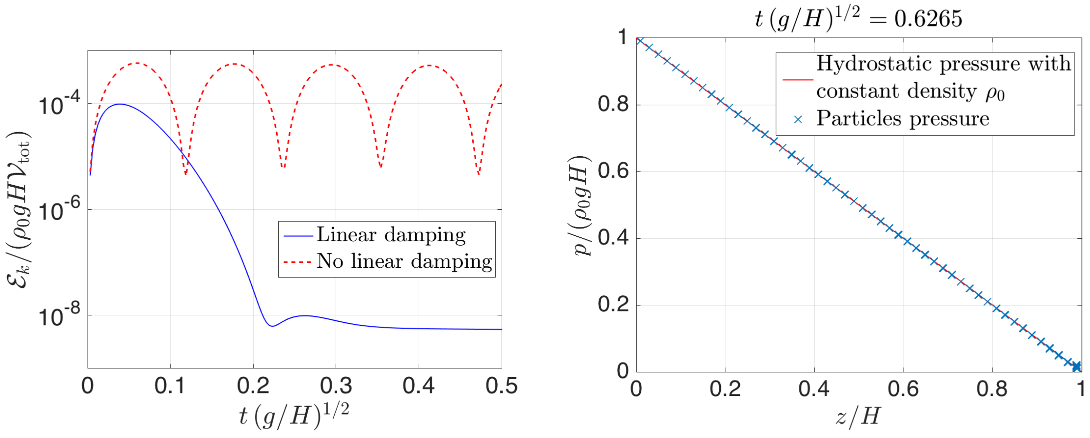


Figure 6.2: Left: Kinetic energy evolution with and without linear damping term; Right: Particles's pressure distribution

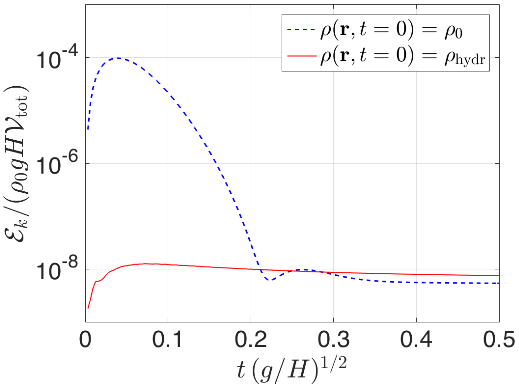


Figure 6.3: Kinetic energy evolution using linear damping under different initial density fields

6.2 HYDROSTATIC EQUILIBRIUM: CONCAVE BOUNDARIES

The second complex geometry presented by Colagrossi et al. (2012) is also tested here. It includes concave boundaries that require the use of the trimming algorithm introduced in Sec. 3.4.3, and allows us to test the dedicated formulation to enforce the solid BCs on them, presented also in Sec. 3.4.3. Table 6.2 summarizes the simulation parameters chosen in this case.

Table 6.2: Simulation parameters: Hydrostatic equilibrium (concave boundaries)

General:	2D,	# of particles: 7427,	$\Delta x_0 = 0.02$ m,	$\boldsymbol{g} = [0, -9.81]$ m/s^2			
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	0.20 s	p_0	0 Pa	Diffusive δ-term?	✓	Average weights?	✗
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	0.02	Initialization ramp?	✓
$\Delta t_{\min}$	10^{-6} s	ν	—	ζ	0.25	t_s	0.20 s
$N_{p\,\max}$	7500	c	50 m/s	Δn_ρ	—	$n_{\max\,s}$	—
$\AE_{\max}$	3	γ	1	CFL	0.20	U	—
						$\mathcal{E}_0$	—

Figure 6.4 shows the different zones in the fluid domain, including a detail of the cell shapes around the concave corner.

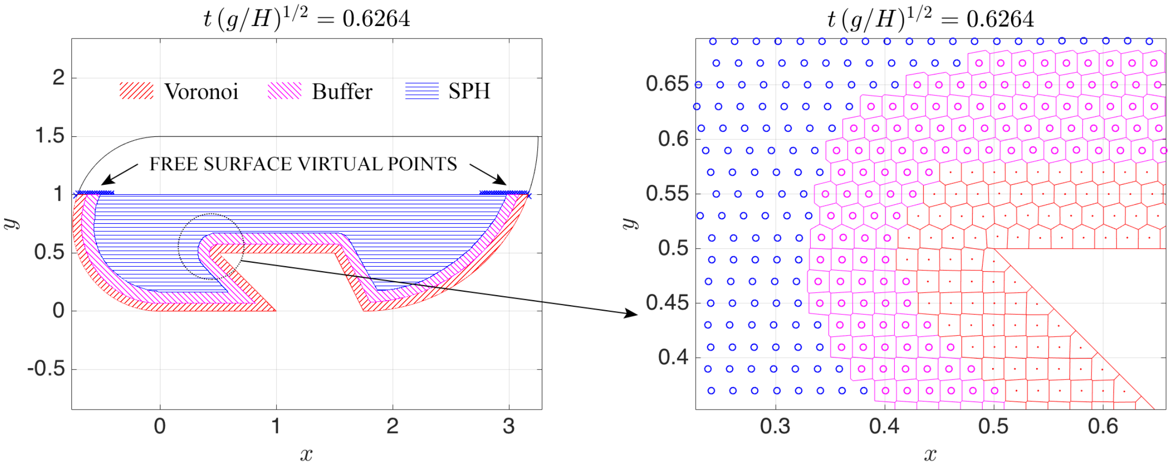


Figure 6.4: Left: Zones in a hydrostatic equilibrium simulation with the complex geometry from Colagrossi et al. (2012). Right: Detail of the Voronoi cell shapes next to the concave vertex.

Figure 6.5 shows the pressure field obtained, zooming near the concave corner on its right panel, and confirming the capability of the current numerical scheme to deal with more complex boundaries. Like in the trapezoidal configuration before, the pressure values from each individual particle show good agreement with the hydrostatic incompressible exact solution, as shown in Fig. 6.6. Finally, Fig. 6.7 shows the initial stabilization scheme performance through the evolution of the kinetic energy, with the density field initialized with $\rho(\boldsymbol{r}, t = 0) = \rho_{\text{hydr}}$.

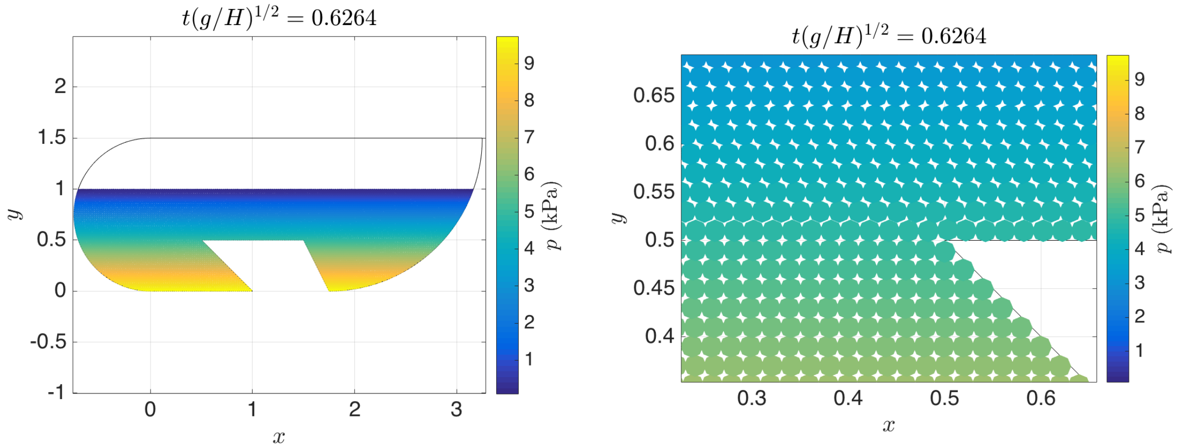


Figure 6.5: Hydrostatic equilibrium with concave boundaries: Pressure field

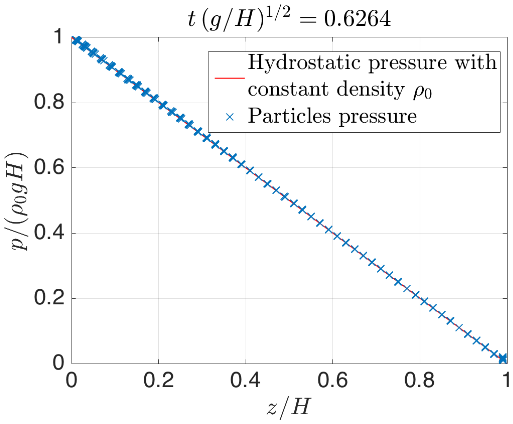


Figure 6.6: Hydrostatic equilibrium with concave boundaries: Particles pressure distribution

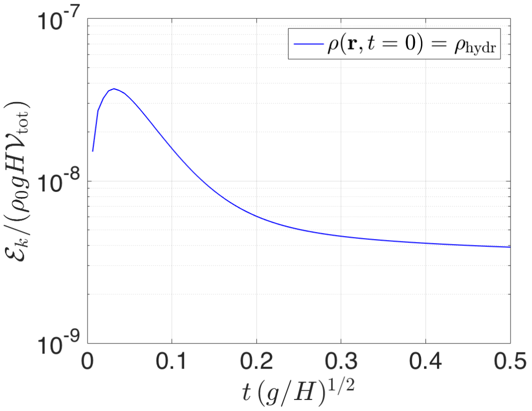


Figure 6.7: Hydrostatic equilibrium with concave boundaries: Kinetic energy evolution

6.3 SOUND WAVE

The propagation and ulterior reflection of a sound wave is discussed next, aiming to investigate how pressure-driven flows are treated by the coupled scheme. The case was first used by Ott and Schnetter (2003) to assess the accuracy of a multiphase SPH implementation. In our case, a rectangular domain of 1×0.03 m, periodic in y, is considered. The particles are evenly distributed and a small amplitude perturbation in their density, $\Delta\rho(x)$, is set according to the following formula:

$$\rho(\boldsymbol{r}, t = 0) = \rho_0 + \Delta\rho(x) = \rho_0 \; + \; A\, \rho_0\, x^2\; \exp\left\{-\left(\frac{x - x_0}{W}\right)^2\right\}, \qquad x \in (0, L) \tag{6.2}$$

where A controls the amplitude of the pressure wave, W controls its width, x_0 is the location of the initial peak of the pressure wave, and L the longitudinal extension of the fluid domain. In this simulation, their values are set to $A = 0.005$ to have density variations of ~1%, $W = 0.1\,L$, $x_0 = 0.5\,L$, and $L = 1$ m from the rectangular domain dimensions. An initial horizontal velocity is given to the particles in order to privilege one propagation direction against the other,

$$\begin{cases} \boldsymbol{u}^x(\boldsymbol{r}, t = 0) = c\,\dfrac{\Delta\rho(x)}{\rho_0}\,, \\ \boldsymbol{u}^y(\boldsymbol{r}, t = 0) = 0\,. \end{cases} \tag{6.3}$$

The flow is treated as inviscid, so the artificial viscosity is included for numerical stability. However, the diffusive δ-correction term is not included, since its purpose is precisely to attenuate the pressure waves that we want to observe. Table 6.3 summarizes the simulation parameters.

Table 6.3: Simulation parameters: Sound wave

General: 2D, # of particles: 8000, $\Delta x_0 = 0.0025$ m, $\boldsymbol{g} = [0, 0]$ m/s^2							
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	0.20 s	p_0	0 Pa	Diffusive δ-term?	✗	Average weights?	✗
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	0.02	Initialization ramp?	✗
$\Delta t_{\min}$	10^{-6} s	ν	—	ζ	0.25		
$N_{p\,\max}$	10000	c	40 m/s	Δn_ρ	—		
Æ$_{\max}$	3	γ	1	CFL	0.20		

A cross section of a 2D sound wave is shown at various time steps in Fig. 6.8, where $t^* = t\,c/L$. Since $x_0 = 0.5\,L$ is taken as the initial, position, $t^* = 1$ corresponds to the moment when the wave returns to the original position after one reflection. The results show how the wave maintains its shape reasonably well after its propagation, and reflection in the Voronoi region adjacent to the lateral walls.

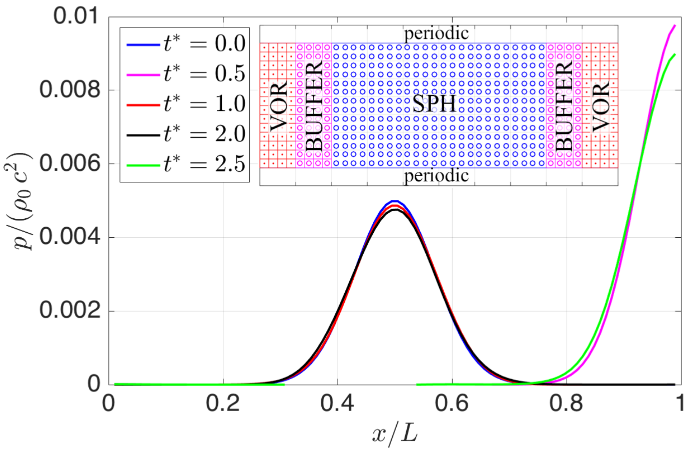


Figure 6.8: 2D sound wave propagation simulation. Pressure profiles at different time steps

Figure 6.9 shows the evolution of the energy components variation as defined in Eq. (5.46):[1] $\widehat{\Delta\mathcal{E}} = (\mathcal{E} - \mathcal{E}^0)/\mathcal{E}^0_{\text{tot}}$. Physically, we can observe the conversion between kinetic and elastic energy in the reflections (as defined in Eqs. (5.26) and (5.28), respectively), with an overall decay due to numerical dissipation.

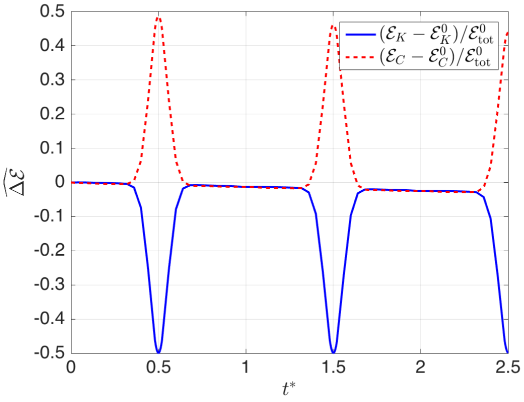


Figure 6.9: Sound wave energy decay

---

[1] Since no thermal nor gravitational effects are included, $\mathcal{E}_P^0 = 0$ and $\mathcal{E}_T^0 = 0$ and therefore $\mathcal{E}_{\text{tot}} = \mathcal{E}_K + \mathcal{E}_C$ in this case

6.4 2D STEADY COUETTE FLOW

A plane Couette viscous flow is considered in this case, shown in Fig. 6.10. The height of the channel is set to $a = 0.1$, with the upper wall moving horizontally with velocity $U = 1$ m/s and the lower wall remaining static. The velocity field is initialized with its exact solution,

$$\begin{cases} \boldsymbol{u}^x(\boldsymbol{r}, t = 0) = U\dfrac{y}{a}\,, \\ \boldsymbol{u}^y(\boldsymbol{r}, t = 0) = 0\,, \end{cases} \tag{6.4}$$

and since no gravitational effects are included, the density field is initialized with $\rho(\boldsymbol{r}, t = 0) = \rho_0$. The viscosity is set to achieve $Re = a\,U/\nu = 10$. Table 6.4 summarizes the simulation parameters.

Table 6.4: Simulation parameters: Steady Couette flow

General: 2D, # of particles: 960, $\Delta x_0 = 0.005$ m, $\boldsymbol{g} = [0, 0]$ m/s^2							
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	0.20 s	p_0	5000 Pa	Diffusive δ-term?	✓	Average weights?	✗
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	—	Initialization ramp?	✗
$\Delta t_{\min}$	10^{-6} s	ν	0.01 m^2/s	ζ	0.25		
$N_{p\,\max}$	1000	c	10 m/s	Δn_ρ	—		
$\mathcal{AR}_{\max}$	3	γ	1	CFL	0.20		

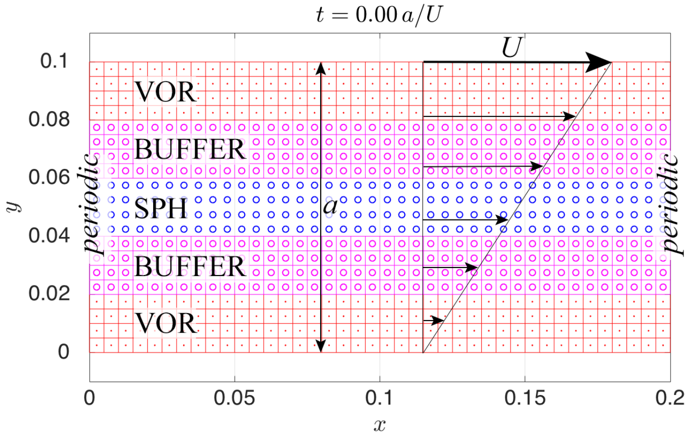


Figure 6.10: 2D steady Couette flow

From its analytic solution we can state that

$$\boldsymbol{u} = \left(U\frac{y}{a}, 0\right) \rightarrow \boldsymbol{r}(t) = \left(x_0 + t\,U\,\frac{y}{a},\, y_0\right),$$

$$\left.\begin{aligned} &\boldsymbol{r}_i(t = 0) = (0, 0), \\ &\boldsymbol{r}_j(t = 0) = (-z, z), \quad z \in [0, a], \end{aligned}\right\} \rightarrow \left.\begin{aligned} &\boldsymbol{r}_i(t = a/U) = (0, 0), \\ &\boldsymbol{r}_j(t = a/U) = (0, z), \end{aligned}\right\},$$

where (x_0, y_0) represent the initial positions. Basically, all the particles located at a -45° diagonal on an initial Cartesian arrangement become vertically aligned after $t = a/U$. Figure 6.11 shows the shape of one Voronoi cell at different time steps during the simulation, showing how the theoretical prediction is achieved and the cells deform due to the different relative velocity recovering the original shape.

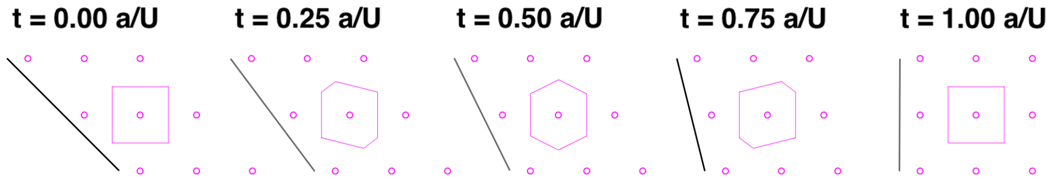


Figure 6.11: Detail of one Voronoi cell at different time steps with the stationary Couette flow.

6.5 2D Impulsively-started Couette flow

The starting Couette flow allows us to test the performance of the code under strong shear flows. While the particle size is kept equal to the steady case before, the height of the channel is enlarged to $a = 1$ m, proportionally increasing the resolution. The fluid is initialized at rest (zero initial velocity), with the upper boundary located at $y = a$ moving horizontally with velocity $U = 1$ m/s in the positive x direction and the lower boundary at $y = 0$ moving with the same velocity but opposite direction, like the case presented by Monaghan (2005a). $\rho(\boldsymbol{r}, t = 0) = \rho_0$ and viscosity is adjusted to achieve $Re = a\,U/\nu = 10$, as shown in Tab. 6.5 and same than the steady case before.

Table 6.5: Simulation parameters: Impulsive starting Couette flow

General:	2D,	# of particles: 5600,	$\Delta x_0 = 0.005$ m,	$\boldsymbol{g} = [0, 0]$ m/s^2			
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	0.20 s	p_0	1250 Pa	Diffusive δ-term?	✓	Average weights?	✗
$n_{\max}$	—	ρ_0	10 kg/m^3	α	—	Initialization ramp?	✗
$\Delta t_{\min}$	10^{-6} s	ν	0.1 m^2/s	ζ	0.25		
$N_{p\,\max}$	6000	c	50 m/s	Δn_ρ	—		
$\AE_{\max}$	3	γ	1	CFL	0.20		

The left panel of Figure 6.12 shows the different zones within the 0.1 × 1 m rectangular fluid domain. Its right panel presents multiple velocity profiles along the vertical direction at various time steps. Time is made non-dimensional using the mean lifetime of the transition to stationary state, such that

$$\tau = t\,\frac{\nu}{\rho_0\,(a/2)^2}\,. \tag{6.5}$$

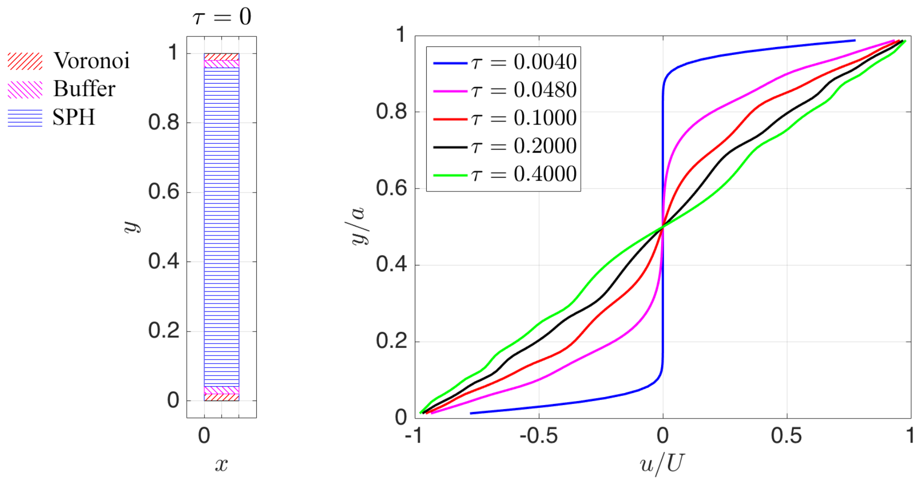


Figure 6.12: Impulsive starting Couette. Left: Fluid zones; Right: velocity profiles at various τ

Figure 6.13 shows the horizontal velocity profile along the vertical direction at a given τ, comparing the numerical results with the theoretical solution given by Batchelor (1967) and adapted to present boundary conditions,

$$\frac{\boldsymbol{u}^{x}(y,t)}{U} = -\left(1-\frac{2y}{a}\right) - \frac{2}{\pi}\sum_{j=1}^{\infty}\frac{1}{j}\sin(j\pi\tilde{y})\exp(-j^2\pi^2\tau)\,, \tag{6.6}$$

where $\tilde{y}$ indicates the distance to the closest boundary made non-dimensional with $a/2$,

$$\tilde{y} = \begin{cases} 2-\frac{y}{a} & y > a\,, \\ \frac{y}{a} & y \leq a\,. \end{cases} \tag{6.7}$$

The agreement is reasonable but small deviations from the exact solution are noticeable.

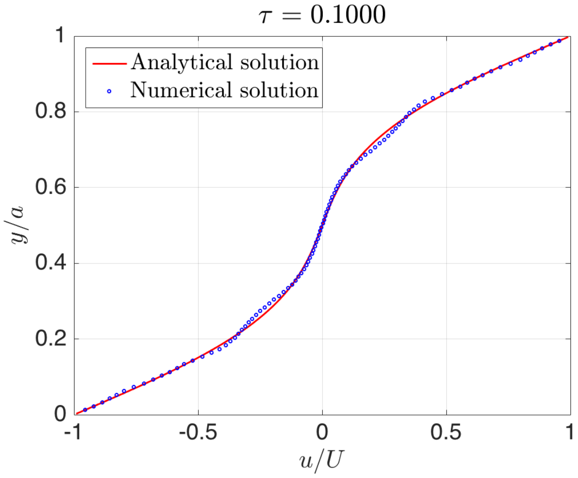


Figure 6.13: Starting Couette velocity profile at $\tau = 0.1$ under $Re = 10$

6.6 LAMB–OSEEN VORTEX

This test case also represents a shear flow, in particular a pure viscous diffusion process. Its analytic solution is presented by Macià et al. (2012), where two types of fluid domains are considered for the numerical simulations: circular and square (Fig. 6.14). The theoretical velocity and density distributions are assigned to the fluid particles as the initial conditions, leaving them free to move afterward under the sole no-slip boundary conditions on the enclosing walls.

$$\rho(\boldsymbol{r}, t = 0) = \rho_0\,, \tag{6.8}$$

$$\boldsymbol{u}(\boldsymbol{r}, t = 0) = (\boldsymbol{u}^x, \boldsymbol{u}^y) = q\,\frac{1 - \exp\left(-\frac{r^2}{a^2}\right)}{r^2}\,(-y, x)\,, \tag{6.9}$$

$$a(t)^2 = a_0^2 + 4\,\nu\,t\,, \tag{6.10}$$

where q gives the intensity of the vortex, a is the effective vortex core radius where the velocity is maximum, ν is the kinematic viscosity, and $r = \|\boldsymbol{r}\| = \|(x, y)\|$ is the magnitude of the position vector. Both circular and square configurations have been simulated using the same input values than Maciá et al.: $q = 0.5$, $a_0 = 1$, and $Re = \frac{2\pi q}{\nu} = 10$. The number of particles differ from each simulation, Table 6.6 summarizes the simulation parameters.

Table 6.6: Simulation parameters: Lamb–Oseen vortex

General:	2D,	# of particles: $\begin{cases} 12172 & \text{square} \\ 15376 & \text{circular} \end{cases}$,	$\Delta x_0 = 0.2$ m,	$\boldsymbol{g} = [0, 0]$ m/s^2		
Stopping criteria		**Fluid properties**		**Numerical scheme**		
$t_{\max}$	10 s	p_0	1250 Pa	Diffusive δ-term? ✓		Average weights?
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	—	circular: ✗
$\Delta t_{\min}$	10^{-6} s	ν	$0.1\,\pi$ m^2/s	ζ	0.25	square: ✓
$N_{p\,\max}$	17000	c	5 m/s	Δn_ρ	—	Initialization ramp? ✗
$AR_{\max}$	3	γ	1	CFL	0.20	

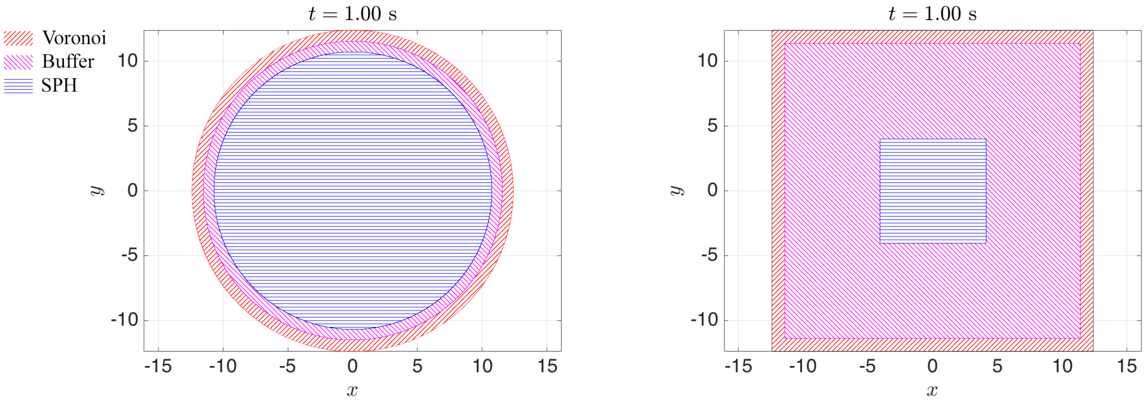


Figure 6.14: Fluid zones in the Lamb–Oseen vortex simulation

Figures 6.15 and 6.16 show the resulting velocity fields, displaying on their right the evolution of the calculated maximum velocity together with the analytic solution. The simulation results show a good agreement with the theoretical values. In addition, we can observe how the conservative formulation in the square configuration, achieved by averaging the coupling weights (Sec. 3.3.2), removes the spurious oscillations visible in the circular one.

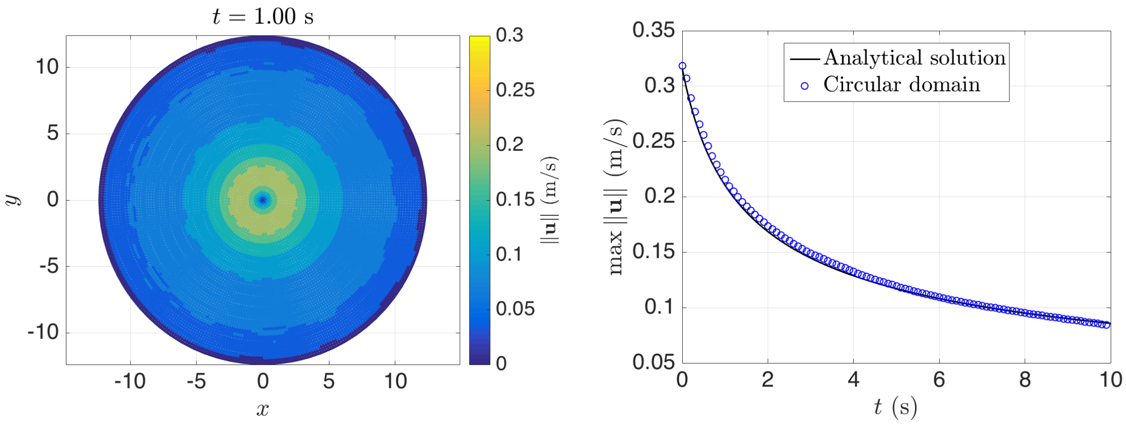


Figure 6.15: Left: Lamb–Oseen vortex velocity fields in a circular domain at $t = 1$ s; Right: Evolution of max $\|\boldsymbol{u}\|$ for the same configuration

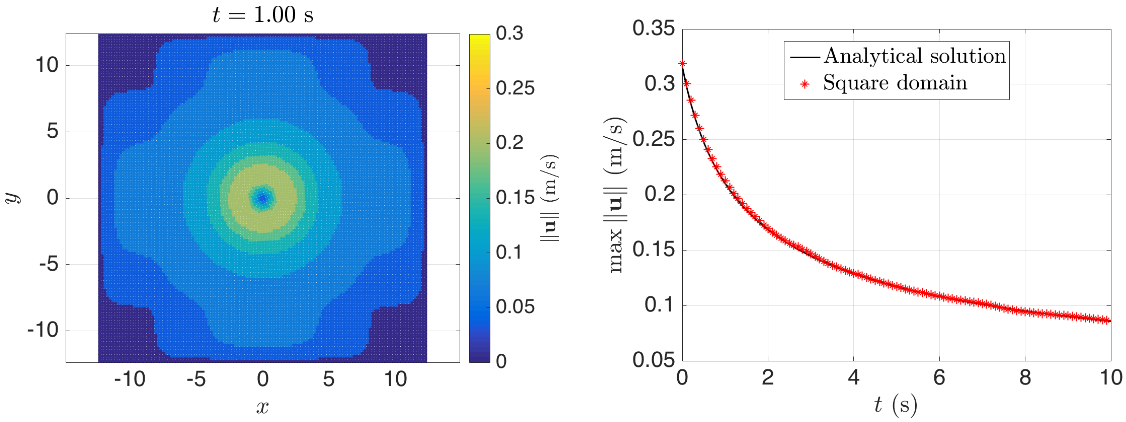


Figure 6.16: Left: Lamb–Oseen vortex velocity fields in a square domain at $t = 1$ s; Right: Evolution of max $\|\boldsymbol{u}\|$ for the same configuration

6.7 Two symmetric Lamb–Oseen vortices

Under the square domain configuration in the previous single Lamb–Oseen case, particles move across the buffer changing their weight values. However, the velocities in this zone are relatively small, making it difficult to extract conclusions. An alternative case with two symmetric Lamb–Oseen vortices has been simulated to visualize better this feature of the coupled scheme, as shown in Fig. 6.17. The coupling weights are averaged like in the square configuration before, and the same values of q, a_0, and ρ are used, although the walls are closer to the vortex centers in this case. The viscous dissipation is reduced by applying the free slip BC at the walls and reducing the viscosity to achieve $Re = \frac{2\pi q}{\nu} = 60$, which facilitates the movement of the particles.

Table 6.7: Simulation parameters: Two symmetric Lamb–Oseen vortices

General: 2D, # of particles: 7688, $\Delta x_0 = 0.1$ m, $\boldsymbol{g} = [0, 0]$ m/s^2

Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	20 s	p_0	1250 Pa	Diffusive δ-term?	✓	Average weights?	✓
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	—	Initialization ramp?	✗
$\Delta t_{\min}$	10^{-6} s	ν	0.05 m^2/s	ζ	0.25		
$N_{p\,\max}$	8000	c	5 m/s	Δn_ρ	20		
$\mathcal{AR}_{\max}$	3	γ	1	CFL	0.20		

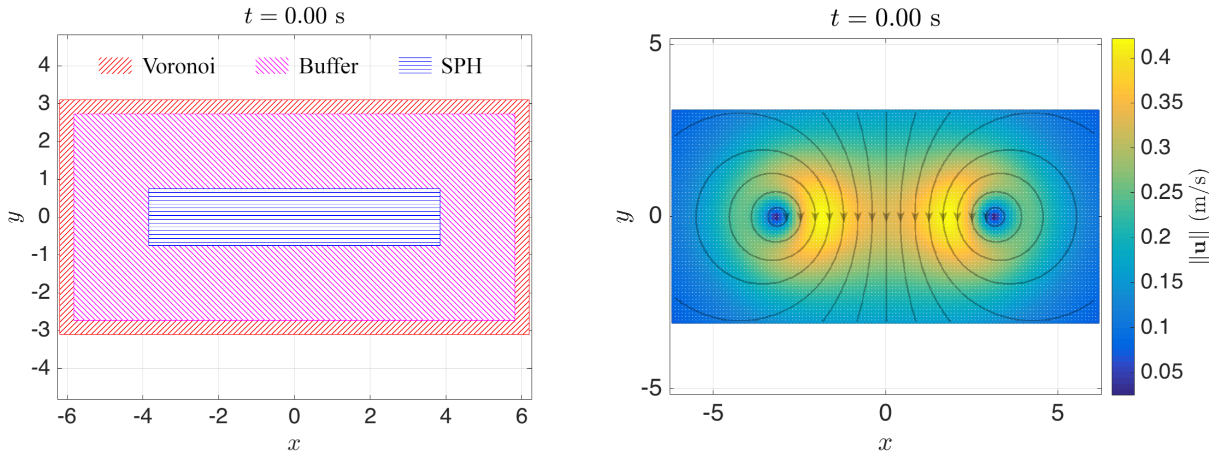


Figure 6.17: Left: Initial fluid zones in the simulation with two symmetric Lamb–Oseen vortices. Right: Initial velocity field for the same configuration, with velocity streamlines superimposed

Figure 6.18 shows the resulting particle distribution at various time steps. Each particle is colored based on its initial vertical position, $y_0 = y(t = 0)$. It can be observed how the mixing induced by the vortices leads to a movement of particles into and away from the walls, crossing smoothly the buffer zone. These results show how the coupled scheme is also able to transport mass across the buffer.

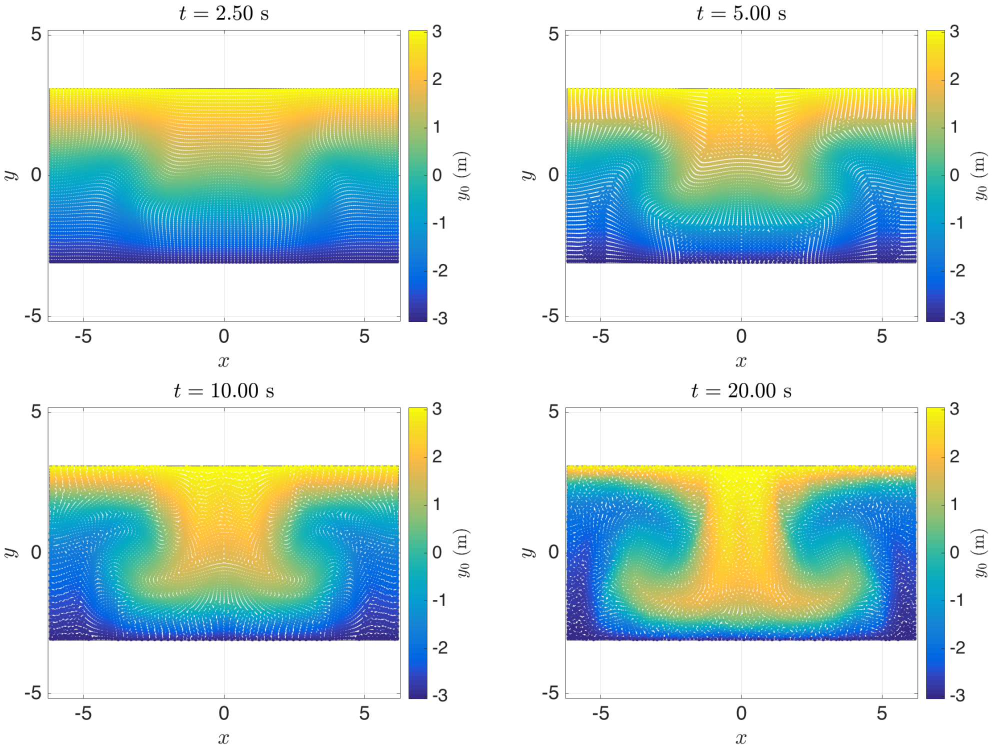


Figure 6.18: Particle configuration at different time steps with the two symmetric Lamb–Oseen vortices. Particles colored with their initial vertical position

6.8 IMPINGING JET

This last verification case involves a velocity inlet, through which fluid particles are introduced in a rectangular computational domain. Table 6.8 summarizes the simulation parameters, and Fig. 6.19 shows the particle distribution at the last time step.

Table 6.8: Simulation parameters: Impinging jet

General:	2D,	# of particles: 2040,	$\Delta x_0 = 0.01$ m,	$\boldsymbol{g} = [0, -9.81]$ m/s^2			
Stopping criteria		**Fluid properties**		**Numerical scheme**			
$t_{\max}$	1 s	p_0	0 Pa	Diffusive δ-term?	✓	Average weights?	✓
$n_{\max}$	—	ρ_0	1000 kg/m^3	α	0.02	Initialization ramp?	✗
$\Delta t_{\min}$	10^{-6} s	ν	—	ζ	0.25		
$N_{p\,\max}$	3000	c	40 m/s	Δn_ρ	20		
$Æ_{\max}$	3	γ	1	CFL	0.20		

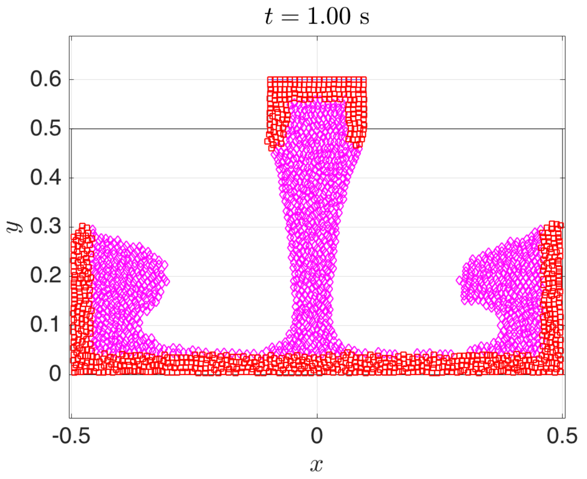


Figure 6.19: Particle distribution during the impinging jet simulation (red squares, Voronoi; magenta diamonds, buffer; no SPH particles exist since the extension of the computational domain is covered entirely by the buffer)

Figure 6.20 shows the particle distribution at various time steps. Initially, the fluid domain is empty and only I/O particles are included. As time goes on, the fluid column falling from the inlet stretches due to gravity before it hits the bottom. After the impact, the fluid expands laterally until it reaches the sides when a second collision with the walls occurs. The fluid moves upwards but is slowed by gravity, and as it runs down it overturns backwards and plunges into the underlying fluid, similar to the dam break case analyzed by Colagrossi and Landrini (2003).

Given that particles under the VPH formulation only interact with their immediate neighbors, they are more sensitive to their relative movement than under the SPH scheme. As a result, they

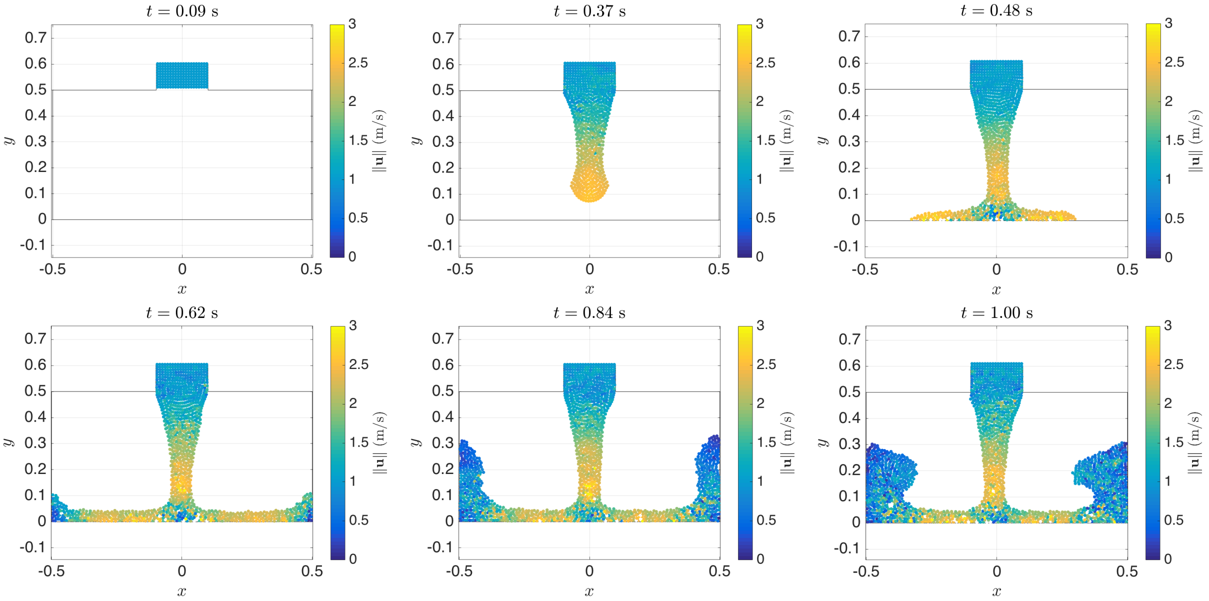


Figure 6.20: Particle configuration at different time steps for the impinging jet simulation. Particles colored with their velocity magnitude.

show a larger cohesion than in SPH, which increases the difficulty in capturing the free-surface deformations. This can be appreciated in Fig. 6.20 both when the fluid is sliding over the bottom and when is overturning, where a narrower shape in the former and a larger detachment in the latter were expected. Therefore, we conclude that a finer resolution will be needed to capture violent free-surface deformations if they occur within the VPH sub-domain.

CHAPTER 7

CONCLUDING REMARKS AND FUTURE WORK

7.1 CONCLUSIONS

The methodology and results presented in this work show the feasibility of combining SPH with Voronoi diagrams into a hybrid Lagrangian SPH-VPH scheme, with an explicit weakly compressible formulation for both methods. While SPH is based on smoothing using a kernel function, VPH uses volume averaged approximations based on a Voronoi tessellation (Serrano and Español, 2001; Hess and Springel, 2010). However, both methods are conceptually very close, making it logical to extend to VPH the stabilization corrections proposed in the literature for SPH. As such, new artificial viscosity and diffusive terms are included in the VPH scheme inspired by the δ-SPH standard formulation (Antuono et al., 2010, 2012; Cercos-Pita et al., 2016).

The fluid domain is subdivided into three separate zones: close to the boundaries where only VPH is applied; far from the boundaries where only SPH is applied; and a transition zone where all fields are computed using both methods and combined through a weighting function. A detailed analysis of the size of the transition (buffer) zone is included to discuss how the conservation properties of both SPH and VPH are kept under this coupling.

Despite the associated computational cost to determine the Voronoi tessellation, it provides additional information beneficial for the numerical scheme. Firstly, the Voronoi tessellation associates a distinct portion of the fluid domain to each particle, *i.e.* its Voronoi cell, which can be used to determine a density value that will conserve volume as well. In addition, this value can be used to re-initialize the density field with a given periodicity, beneficial to reduce the high-frequency pressure oscillations. In this way, it corrects the numerical inconsistencies derived from evolving the density by integrating the continuity equation, while it maintains the benefits of including the δ-correction to mitigate the undesired sonic effects. The clustering of particles is also avoided by limiting the deviation of the particle position from the centroid of its cell through a shifting algorithm inspired by Lloyd (1982).

However, the main advantage of VPH lies on the enforcement of the boundary conditions. Under the VPH scheme, the interaction with neighbor elements is done through the cell face shared with them. By overlapping the boundaries with cell faces:

- no cell crosses the boundary, so we maintain the consistency of the volumetric subdivision of the fluid domain, and
- we use the known physical values at the face from the boundary conditions to evaluate the VPH differential operators. Not only it avoids requiring (ghost) fluid particles beyond the boundary, but it is also directly applicable in 3D and higher dimensions with no further modification of the scheme.

Special emphasis has been placed in the algorithm generating the tessellation to enforce this exact mapping of boundary faces. Moreover, it is done without imposing any restriction on the particles movement. This is achieved by using the convex boundaries as symmetry planes to reflect the particles next to them, whereas concave boundaries are used to trim the unbounded cells crossing them. In addition, the boundaries themselves must be discretized into a set of planar elements (segments in 2D, triangles in 3D).

The inclusion of free-surfaces in VPH represented a challenge, with no previous attempts found in the literature. In this work, a new algorithm to detect angular regions around particles not covered by neighbors is used to detect the free-surface location. Periodic and inlet/outlet boundaries complete the possibilities to surround the fluid domain, which are treated by including an additional layer of Voronoi particles beyond the boundary to ensure that the fluid particles are properly bounded.

In summary, the new coupled scheme proposed extends the artificial viscosity and δ-correction to the VPH formulation, incorporates free-surfaces to it for the first time, and is capable of dealing with complex geometries. The treatment of the boundary conditions is greatly simplified by using Voronoi diagrams, compensating for the additional computational cost that it requires building the tessellation.

The method has been coded in a hybrid Matlab–C++ implementation, using a first-order leap frog time-stepping scheme. During the program initialization, a linear damping term is introduced to remove the initial instabilities from unbalanced initial conditions following Monaghan (1994). The energy balance at the discrete level of the scheme is analyzed, identifying its different components.

The accuracy of the coupled scheme is discussed by means of a set of well-known verification benchmarks. The results showed that pressure gradient dominated problems, such as hydrostatic conditions or sound wave propagations, are well simulated by the method. However, problems which are dominated by viscous diffusion render less accurate results, consistent with the increased error made in the higher order differential operators. The oscillations observed in the impulsively-started Couette reflect the impact of sonic effects, even in shear driven fluids. While they progressively vanish, a larger attenuation or a lower sensitivity to density variations must be considered if the transient phase is the main focus of the simulation.

7.2 RECOMMENDATIONS FOR FUTURE WORK

While this work establishes the foundation of the δ-VSPH method, it remains to be further developed in multiple areas:

- Parallelize the code to be able to run in multiple processors through external libraries such *OpenMP*[1] or *OpenMPI*[2]. This will allow us to use larger number of particles and tackle more complex geometries and flows.
- Validate the method with experimental results (Zhou et al., 1999; Lobovský et al., 2014).
- Run simulations with 3D configurations and moving bodies to demonstrate the full potential of the method, more feasible once the code is parallelized.

---

[1] `http://www.openmp.org`

[2] `https://www.open-mpi.org`

- Analyze its consistency under large free-surface deformations with air entrapment (Colagrossi and Landrini, 2003), and explore more advanced free-surface detection methods proposed for SPH (Dilts, 2000; Haque and Dilts, 2007; Marrone et al., 2010; Marrone, 2011).
- Include thermal effects, multi-phase flows, and surface-tension loads in the simulations.
- Reduce the computational demands and/or improve the precision of the method by including variable time scales for different locations of the flow (multirate methods), higher order time-stepping algorithms like RK4, adaptive artificial viscosity to model shocks (Balsara, 1995), and particle merging/splitting algorithms for variable spatial resolution (Shapiro et al., 1996; Lastiwka et al., 2005; Vacondio et al., 2013; Hu et al., 2017).

The great advantage of the coupled scheme presented is that it combines the core formulations of SPH and VPH, but does not establish any further restriction on each. As such, incorporating the existing modifications in the literature should be straightforward, a priori. Once again, the large similarities between both methods will simplify transferring the techniques proposed for the more mature SPH to VPH, anticipating an exciting future for this coupled approach.

7.3 ASSOCIATED PUBLICATIONS

- D. Fernandez-Gutierrez, A. Souto-Iglesias, T. I. Zohdi (11/2017) “A hybrid Lagrangian Voronoi-SPH scheme”, *Journal of Computational Particle Mechanics*, ISSN 2196-4378. URL `https://link.springer.com/article/10.1007%2Fs40571-017-0173-4`
- D. Fernandez-Gutierrez, and T.I. Zohdi, (11/2017). “Coupling SPH with Voronoi diagrams to model solid boundaries”. *70th Annual Meeting of the American Physical Society Division of Fluid Dynamics (APS-DFD)*. Denver, USA.
- D. Fernandez-Gutierrez, A. Souto-Iglesias, T. I. Zohdi (06/2017) “A hybrid Lagrangian Voronoi-SPH scheme”, *12th International SPHERIC SPH Workshop*. Ourense, Spain.

REFERENCES

Adami, S., Hu, X., and Adams, N. (2012). "Simulating three-dimensional turbulence with SPH". In *Proceedings of the Summer Program*, p. 177. Center for Turbulence Research.

Allen, M. P. and Tildesley, D. J. (2017). *Computer simulation of liquids*. Oxford university press.

Antoci, C., Gallati, M., and Sibilla, S. (2007). "Numerical simulation of fluid–structure interaction by SPH". *Computers & Structures*, 85(11):pp. 879 – 890. ISSN 0045-7949. Fourth MIT Conference on Computational Fluid and Solid Mechanics, URL `http://www.sciencedirect.com/science/article/pii/S0045794907000132`.

Antuono, M., Colagrossi, A., and Marrone, S. (2012). "Numerical diffusive terms in weakly-compressible SPH schemes". *Computer Physics Communications*, 183(12):pp. 2570–2580.

Antuono, M., Colagrossi, A., Marrone, S., and Molteni, D. (2010). "Free-surface flows solved by means of SPH schemes with numerical diffusive terms". *Computer Physics Communications*, 181(3):pp. 532–549. ISSN 0010-4655. URL `http://www.sciencedirect.com/science/article/pii/S0010465509003506`.

Antuono, M., Marrone, S., Colagrossi, A., and Bouscasse, B. (2015). "Energy balance in the δ-SPH scheme". *Computer Methods in Applied Mechanics and Engineering*, 289:pp. 209–226.

Aurenhammer, F. (1987). "Power diagrams: Properties, algorithms and applications". *SIAM Journal on Computing*, 16(1):pp. 78–96.

Balsara, D. S. (1995). "Von Neumann stability analysis of Smoothed Particle Hydrodynamics - Suggestions for optimal algorithms". *Journal of Computational Physics*, 121:pp. 357–372.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). "The quickhull algorithm for convex hulls". *ACM Trans. Math. Softw.*, 22(4):pp. 469–483. ISSN 0098-3500. URL `http://doi.acm.org/10.1145/235815.235821`.

Barcarolo, D. A. (2013). *Improvement of the precision and the efficiency of the SPH method: theoretical and numerical study*. Ph.D. thesis, Ecole Centrale de Nantes.

Barcarolo, D. A., Touzé, D. L., Oger, G., and de Vuyst, F. (2014). "Voronoi-SPH: on the analysis of a hybrid Finite Volumes - Smoothed Particle Hydrodynamics method". In *9th Int. SPHERIC workshop*, pp. 371–378.

Barreiro, A., Domínguez, J. M., Crespo, A. J. C., García-Feal, O., and Gesteira, M. G. (2015). "Smoothed Particle Hydrodynamics for free-surface flows". In J. Klapp, G. Ruíz Chavarría, A. Medina Ovando, A. López Villa, and L. D. G. Sigalotti, eds., *Selected Topics of Computational and Experimental Fluid Mechanics*, pp. 119–136. Springer International Publishing, Cham. ISBN 978-3-319-11487-3. URL `https://doi.org/10.1007/978-3-319-11487-3_6`.

Batchelor, G. K. (1967). *Introduction to Fluid Dynamics*. Cambridge University Press, New York.

Becker, M. and Teschner, M. (2007). "Weakly compressible SPH for free surface flows". In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 209–217. Eurographics Association.

Belytschko, T., Krongauz, Y., Dolbow, J., and Gerlach, C. (1998). "On the completeness of meshfree particle methods". *Int. J. Numer. Methods Engineering*, 43(5):pp. 785–819.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P. (1996). "Meshless methods: An overview and recent developments". *Computer Methods in Applied Mechanics and Engineerings*, 139:pp. 3–47.

Benz, W. (1988). "Applications of Smoothed Particle Hydrodynamics (SPH) to Astrophysical Problems". *Computer Physics Communications*, 48:pp. 97–105.

Benz, W. and Asphaug, E. (1994). "Impact Simulations with Fracture. I. Method and Tests". *Icarus*, 107(1):pp. 98–116. ISSN 0019-1035. URL http://www.sciencedirect.com/science/article/pii/S0019103584710098.

Benz, W. and Asphaug, E. (1995). "Simulations of brittle solids using smooth particle hydrodynamics". *Computer Physics Communications*, 87(1–2):pp. 253–265.

Bicknell, G. V. and Gingold, R. A. (1983). "On tidal detonation of stars by massive black holes". *The Astrophysical Journal*, 273:pp. 749–760.

Bird, G. A. (1976). "Molecular gas dynamics". *NASA STI/Recon Technical Report A*, 76.

Bonet, J. and Lok, T. (1999). "Variational and momentum preservation aspects of Smoothed Particle Hydrodynamics formulations". *Comput. Methods Appl. Mech. Eng.*, 180:pp. 97–115.

Bowyer, A. (1981). "Computing dirichlet tessellations*". *The Computer Journal*, 24(2):pp. 162–166. /oup/backfile/content_public/journal/comjnl/24/2/10.1093/comjnl/24.2.162/2/240162.pdf.

Cercos-Pita, J. (2015). "AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL". *Computer Physics Communications*, 192(0):pp. 295–312. ISSN 0010-4655. URL http://www.sciencedirect.com/science/article/pii/S0010465515000909.

Cercos-Pita, J., Antuono, M., Colagrossi, A., and Souto-Iglesias, A. (2017). "SPH energy conservation for fluid–solid interactions". *Computer Methods in Applied Mechanics and Engineering*, 317(Supplement C):pp. 771–791. ISSN 0045-7825. URL http://www.sciencedirect.com/science/article/pii/S0045782516303644.

Cercos-Pita, J., Dalrymple, R., and Herault, A. (2016). "Diffusive terms for the conservation of mass equation in SPH". *Applied Mathematical Modelling*, 40(19):pp. 8722 – 8736. ISSN 0307-904X. URL http://www.sciencedirect.com/science/article/pii/S0307904X16302669.

Cercos-Pita, J. L. (2016). *A novel generalized diffusive SPH model: Theoretical analysis and 3D HPC implementation*. Ph.D. thesis, Technical University of Madrid (UPM).

Chen, S. and Doolen, G. D. (1998). “Lattice Boltzmann method for fluid flows”. *Annual Review of Fluid Mechanics*, 30(1):pp. 329–364. URL https://doi.org/10.1146/annurev.fluid.30.1.329.

Chiaki, G. and Yoshida, N. (2015). “Particle splitting in smoothed particle hydrodynamics based on Voronoi diagram”. *Monthly Notices of the Royal Astronomical Society*, 451(4):pp. 3955–3963.

Chorin, A. J. (1973). “Numerical study of slightly viscous flow”. *Journal of Fluid Mechanics*, 57(4):p. 785?796.

Chorin, A. J. and Bernard, P. S. (1973). “Discretization of a vortex sheet, with an example of roll-up”. *Journal of Computational Physics*, 13(3):pp. 423 – 429. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/0021999173900454.

Cignoni, P., Montani, C., and Scopigno, R. (1998). “DeWall: A fast divide and conquer Delaunay triangulation algorithm in E^d”. *Computer-Aided Design*, 30(5):pp. 333 – 341. ISSN 0010-4485. URL http://www.sciencedirect.com/science/article/pii/S0010448597000821.

Cleary, P. (1996). “New implementation of viscosity. Tests with Couette flows”. Tech. rep., CSIRO Division of Mathematics and Statistics.

Colagrossi, A. (2005). *A meshless lagrangian method for free-surface and interface flows with fragmentation*. Ph.D. thesis, Università di Roma La Sapienza.

Colagrossi, A., Antuono, M., and Le Touzé, D. (2009). “Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model”. *Phys. Rev. E*, 79:p. 056701. URL https://link.aps.org/doi/10.1103/PhysRevE.79.056701.

Colagrossi, A., Antuono, M., Souto-Iglesias, A., and Le Touzé, D. (2011). “Theoretical analysis and numerical verification of the consistency of viscous smoothed-particle-hydrodynamics formulations in simulating free-surface flows”. *Physical Review E*, 84:p. 026705. URL https://link.aps.org/doi/10.1103/PhysRevE.84.026705.

Colagrossi, A., Antuono, M., Souto-Iglesias, A., Le Touzé, D., and Izaguirre-Alza, P. (2010). “Theoretical analysis of SPH in simulating free-surface viscous flows”. In *5th ERCOFTAC SPHERIC workshop on SPH applications*.

Colagrossi, A., Bouscasse, B., Antuono, M., and Marrone, S. (2012). “Particle packing algorithm for SPH schemes”. *Computer Physics Communications*, 183(2):pp. 1641–1683.

Colagrossi, A. and Landrini, M. (2003). “Numerical Simulation of Interfacial Flows by Smoothed Particle Hydrodynamics”. *J. Comp. Phys.*, 191:pp. 448–475.

Colagrossi, A., Souto-Iglesias, A., Antuono, M., and Marrone, S. (2013). “Smoothed-particle-hydrodynamics modeling of dissipation mechanisms in gravity waves”. *Phys. Rev. E*, 87:p. 023302.

Crespo, A., Dominguez, J., Rogers, B., Gomez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A., and Garcia-Feal, O. (2015). "DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)". *Computer Physics Communications*, 187:pp. 204–216. ISSN 0010-4655.

Cummins, S. and Rudman, M. (1999). "An SPH projection method". *J. Comp. Phys.*, 152(2):pp. 584–607.

Das, R. and Cleary, P. (2010). "Effect of rock shapes on brittle fracture using Smoothed Particle Hydrodynamics". *Theoretical and Applied Fracture Mechanics*, 53(1):pp. 47 – 60. ISSN 0167-8442. URL `http://www.sciencedirect.com/science/article/pii/S0167844209001219`.

De Berg, M., Cheong, O., Van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Introduction*. Springer.

De Leffe, M., Le Touzé, D., and Alessandrini, B. (2009). "Normal flux method at the boundary for SPH". In 4^{th} *SPHERIC*, pp. 149–156.

De Leffe, M., Le Touzé, D., and Alessandrini, B. (2011). "A modified no-slip condition in weakly-compressible SPH". In *6th ERCOFTAC SPHERIC workshop on SPH applications*, pp. 291–297.

Di Lisio, R., Grenier, E., and Pulvirenti, M. (1997). "On the regularization of the pressure field in compressible euler equations". *Annali della Scuola Normale Superiore di Pisa. Classe di Scienze. Serie IV*, 24(2):pp. 227–238. ISSN 0391-173X. URL `http://www.numdam.org/item?id=ASNSP_1997_4_24_2_227_0`.

Di Lisio, R., Grenier, E., and Pulvirenti, M. (1998). "The convergence of the SPH method". *Computers & Mathematics with Applications*, 35(1):pp. 95–102. ISSN 0898-1221. URL `http://www.sciencedirect.com/science/article/pii/S0898122197002605`.

Dilts, G. A. (1999). "Moving-least-squares-particle hydrodynamics? I. Consistency and stability". *International Journal for Numerical Methods in Engineering*, 44(8):pp. 1115–1155. ISSN 1115–1155.

Dilts, G. A. (2000). "Moving least-squares particle hydrodynamics II: conservation and boundaries". *International Journal for Numerical Methods in Engineering*, 48(10):pp. 1503–1524. ISSN 1097-0207.

Drazin, P. G. and Riley, N. (2006). *The Navier-Stokes equations: a classification of flows and exact solutions*. 334. Cambridge University Press. ISBN 9780521681629.

Duque, D., Español, P., and de la Torre, J. A. (2017). "Extending linear finite elements to quadratic precision on arbitrary meshes". *Applied Mathematics and Computation*, 301:pp. 201 – 213. ISSN 0096-3003. URL `http://www.sciencedirect.com/science/article/pii/S0096300316307366`.

Español, P. and Revenga, M. (2003). “Smoothed dissipative particle dynamics”. *Phys. Rev. E*, 67:p. 026705. URL `https://link.aps.org/doi/10.1103/PhysRevE.67.026705`.

Español, P. and Warren, P. (1995). “Statistical mechanics of dissipative particle dynamics”. *EPL (Europhysics Letters)*, 30(4):p. 191. URL `http://stacks.iop.org/0295-5075/30/i=4/a=001`.

Español, P. and Warren, P. B. (2017). “Perspective: Dissipative particle dynamics”. *The Journal of Chemical Physics*, 146(15):p. 150901.

Evans, M. W. and Harlow, F. H. (1957). “The particle-in-cell method for hydrodynamic calculations”. Tech. rep., LOS ALAMOS NATIONAL LAB NM. Los Alamos Scientific Laboratory report LA-2139.

Federico, I., Marrone, S., Colagrossi, A., Aristodemo, F., and Antuono, M. (2012). “Simulating 2D open-channel flows through an SPH model”. *European Journal of Mechanics - B/Fluids*, 34:pp. 35 – 46. ISSN 0997-7546. URL `http://www.sciencedirect.com/science/article/pii/S0997754612000337`.

Feldman, J. and Bonet, J. (2007). “Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems”. *International Journal for Numerical Methods in Engineering*, 72(3):pp. 295–324. URL `http://www3.interscience.wiley.com/cgi-bin/abstract/114179937/ABSTRACT`.

Fernández-Gutiérrez, D., Moreu, J., de Guzmán, S., Yeung, R. W., and Moreu, M. (2014). “Numerical simulations in the design of a new Tension-Tethered marine current turbine”. In *Proceedings of the 33rd ASME International Conference on Ocean, Offshore, and Arctic Engineering*.

Ferrari, A., Dumbser, M., Toro, E. F., and Armanini, A. (2009). “A new 3D parallel SPH scheme for free surface flows”. *Computers & Fluids*, 38(6):pp. 1203–1217. ISSN 0045-7930. URL `http://www.sciencedirect.com/science/article/pii/S0045793008002284`.

Flekkøy, E. G., Coveney, P. V., and De Fabritiis, G. (2000). “Foundations of dissipative particle dynamics”. *Phys. Rev. E*, 62:pp. 2140–2157. URL `https://link.aps.org/doi/10.1103/PhysRevE.62.2140`.

Fortune, S. (1987). “A sweepline algorithm for Voronoi diagrams”. *Algorithmica*, 2(1):p. 153. ISSN 1432-0541.

Ghaffari, M. A. and Xiao, S. (2016). “Smoothed Particle Hydrodynamics with Stress Points and Centroid Voronoi Tessellation (CVT) Topology Optimization”. *International Journal of Computational Methods*, 13(05):p. 1650031. URL `http://www.worldscientific.com/doi/abs/10.1142/S0219876216500316`.

Gingold, R. and Monaghan, J. (1977). “Smoothed Particle Hydrodynamics: theory and application to non-spherical stars”. *Mon. Not. Roy. Astron. Soc. (MNRAS)*, 181:pp. 375–389.

Gingold, R. and Monaghan, J. (1982). “Kernel estimates as a basis for general particle methods in hydrodynamics”. *Journal of Computational Physics*, 46(3):pp. 429–453. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/0021999182900250.

González, R. (2016). “PARAVT: Parallel Voronoi tessellation code”. *Astronomy and Computing*, 17(Supplement C):pp. 80 – 85. ISSN 2213-1337. URL http://www.sciencedirect.com/science/article/pii/S2213133716300609.

Gray, J. (2001). *Caldera collapse and the generation of waves*. Ph.D. thesis, Monash University.

Gray, J. and Monaghan, J. (2004). “Numerical modelling of stress fields and fracture around magma chambers”. *Journal of Volcanology and Geothermal Research*, 135(3):pp. 259–283. ISSN 0377-0273. URL http://www.sciencedirect.com/science/article/pii/S0377027304000678.

Haque, A. and Dilts, G. A. (2007). “Three-dimensional boundary detection for particle methods”. *Journal of Computational Physics*, 226(2):pp. 1710–1730. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/S0021999107002586.

Hernquist, L. and Katz, N. (1989). “TreeSPH: A Unification of SPH with the Hierarchical Tree Method”. *Astrophysical Journal Supplement*, 70:pp. 419–446.

Hess, S. and Springel, V. (2010). “Particle hydrodynamics with tessellation techniques”. *Monthly Notices of the Royal Astronomical Society*, 406(4):pp. 2289–2311.

Hieber, S. and Koumoutsakos, P. (2008). “An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers”. *Journal of Computational Physics*, 227(19):pp. 8636–8654. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/S0021999108003343.

Hirsch, C. (1990). *Numerical computation of internal and external flows. Vol. 2: computational methods for inviscid and viscous flows*. John Wiley & Sons.

Hirt, C., Amsden, A., and Cook, J. (1997). “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”. *J. Comput. Phys.*, 135(2):pp. 203–216. ISSN 0021-9991. Reprinted from J. Comput. Phys. 1974; 14:227–253, URL http://dx.doi.org/10.1006/jcph.1997.5702.

Hirt, C. and Nichols, B. (1981). “Volume of fluid (VOF) method for the dynamics of free boundaries”. *Journal of Computational Physics*, 39(1):pp. 201 – 225. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/0021999181901455.

Hoover, W. (1998). “Isomorphism linking smooth particles and embedded atoms”. *Physica A: Statistical Mechanics and its Applications*, 260(3):pp. 244 – 254. ISSN 0378-4371. URL http://www.sciencedirect.com/science/article/pii/S0378437198003574.

Hu, W., Pan, W., Rakhsha, M., Tian, Q., Hu, H., and Negrut, D. (2017). “A consistent multi-resolution smoothed particle hydrodynamics method”. *Computer Methods in Applied Mechanics and Engineering*, 324:pp. 278 – 299. ISSN 0045-7825. URL http://www.sciencedirect.com/science/article/pii/S0045782517305315.

Hu, X. Y. and Adams, N. A. (2006). “Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows”. *Physics of Fluids*, 18(10):p. 101702.

Idelsohn, S., Oñate, E., and Pin, F. D. (2004). “The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves”. *International Journal for Numerical Methods in Engineering*, 61(7):pp. 964–989. ISSN 1097-0207.

Kiara, A. (2010). *Analysis of the smoothed particle hydrodynamics method for free-surface flows*. Ph.D. thesis, Massachusetts Institute of Technology.

Kumar, P., Yang, Q., Jones, V., and McCue-Weil, L. (2015). “Coupled SPH-FVM Simulation within the OpenFOAM Framework”. *Procedia IUTAM*, 18:pp. 76–84. ISSN 2210-9838. URL http://www.sciencedirect.com/science/article/pii/S2210983815002771.

Kundu, P. K. and Cohen, I. (2008). *Fluid mechanics*. Elsevier Acad. Press, London, 4^{th} ed.

Lancaster, P. and Salkauskas, K. (1981). “Surfaces generated by moving least squares methods”. *Mathematics of computation*, 37(155):pp. 141–158.

Lastiwka, M., Quinlan, N., and Basa, M. (2005). “Adaptive particle distribution for smoothed particle hydrodynamics”. *International Journal for Numerical Methods in Fluids*, 47(10-11):pp. 1403–1409. ISSN 1097-0363. URL http://dx.doi.org/10.1002/fld.891.

Leonard, A. (1980). “Vortex methods for flow simulation”. *Journal of Computational Physics*, 37(3):pp. 289 – 335. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/0021999180900406.

Li, S. and Liu, W. K. (2004). *Meshfree Particle Methods*. Springer-Verlag Berlin Heidelberg, 1 ed. ISBN 978-3-540-71471-2.

Libersky, L., Petschek, A., Carney, T., Hipp, J., and Allahdadi, F. (1993). “High strain Lagrangian hydrodynamics a three-dimensional SPH code for dynamic material response”. *Journal of Computational Physics*, 109(1):pp. 67–75.

Libersky, L. D. and Petschek, A. G. (1991). “Smooth particle hydrodynamics with strength of materials”. In H. E. Trease, M. F. Fritts, and W. P. Crowley, eds., *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method: Proceedings of the Next Free-Lagrange Conference Held at Jackson Lake Lodge, Moran, WY, USA 3–7 June 1990*, pp. 248–257. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-46608-6. URL https://doi.org/10.1007/3-540-54960-9_58.

Lind, S., Xu, R., Stansby, P., and Rogers, B. (2012). “Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves”. *Journal of Computational Physics*, 231(4):pp. 1499 – 1523. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/S0021999111006279.

Liu, G. and Liu, M. (2003). *Smoothed particle hydrodynamics. A meshfree particle method*. World Scientific Publishing Co. Pte. Ltd. ISBN 978-981-238-456-0.

Lloyd, S. (1982). "Least squares quantization in PCM". *IEEE Transactions on Information Theory*, 28(2):pp. 129–137. ISSN 0018-9448.

Lobovský, L., Botia-Vera, E., Castellana, F., Mas-Soler, J., and Souto-Iglesias, A. (2014). "Experimental investigation of dynamic pressure loads during dam break". *Journal of Fluids and Structures*, 48(Supplement C):pp. 407–434. ISSN 0889-9746. URL `http://www.sciencedirect.com/science/article/pii/S0889974614000656`.

Lucy, L. (1977). "A numerical approach to the testing of the fission hypothesis". *Astronomical Journal*, 82:pp. 1013–1024.

Macià, F., Antuono, M., González, L. M., and Colagrossi, A. (2011a). "Theoretical Analysis of the No-Slip Boundary Condition Enforcement in SPH Methods". *Progress of Theoretical Physics*, 125(6):pp. 1091–1121. URL `http://ptp.ipap.jp/link?PTP/125/1091/`.

Macià, F., Colagrossi, A., Antuono, M., and Souto-Iglesias, A. (2011b). "Benefits of using a Wendland kernel for free-surface flows". In *6th ERCOFTAC SPHERIC workshop on SPH applications*.

Macià, F., Sánchez, J. M., Souto-Iglesias, A., and González, L. M. (2012). "WCSPH viscosity diffusion processes in vortex flows". *International Journal for Numerical Methods in Fluids*, 69(3):pp. 509–533. ISSN 1097-0363.

Marongiu, J., Leboeuf, F., and Parkinson, E. (2008). "Riemann solvers and efficient boundary treatments: an hybrid SPH-finite volume numerical method". In *3rd ERCOFTAC SPHERIC workshop on SPH applications*. EPFL.

Marongiu, J.-C., Leboeuf, F., Caro, J., and Parkinson, E. (2010). "Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method". *Journal of Hydraulic Research*, 48(sup1):pp. 40–49.

Marrone, S. (2011). *Enhanced SPH modeling of free-surface flows with large deformations*. Ph.D. thesis, PhD thesis, University of Rome, La Sapienza.

Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touzé, D., and Graziani, G. (2011). "Delta-SPH model for simulating violent impact flows". *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):pp. 1526–1542. ISSN 0045-7825.

Marrone, S., Colagrossi, A., Antuono, M., Colicchio, G., and Graziani, G. (2013). "An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers". *Journal of Computational Physics*, 245:pp. 456–475. ISSN 0021-9991.

Marrone, S., Colagrossi, A., Le Touzé, D., and Graziani, G. (2010). "Fast free-surface detection and level-set function definition in SPH solvers". *Journal of Computational Physics*, 229(10):pp. 3652–3663. ISSN 0021-9991. URL `http://www.sciencedirect.com/science/article/pii/S0021999110000343`.

Marrone, S., Mascio, A. D., and Le Touzé, D. (2016). "Coupling of Smoothed Particle Hydrodynamics with Finite Volume method for free-surface flows". *Journal of Computational Physics*, 310:pp. 161–180. Available online 11 December 2015.

Martel, H., Shapiro, P. R., Villumsen, J. V., and Kang, H. (1994). "Adaptive Smoothed Particle Hydrodynamics with Application to Galaxy and Large-Scale Structure Formation (Invited paper)". *Memorie della Società Astronomia Italiana*, 65:p. 1061.

Melenk, J. and Babuška, I. (1996). "The partition of unity finite element method: Basic theory and applications". *Computer Methods in Applied Mechanics and Engineering*, 139(1):pp. 289 – 314. ISSN 0045-7825. URL http://www.sciencedirect.com/science/article/pii/S0045782596010870.

Molteni, D. and Colagrossi, A. (2009). "A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH". *Computer Physics Communications*, 180(6):pp. 861–872. ISSN 0010-4655. URL http://www.sciencedirect.com/science/article/pii/S0010465508004219.

Monaghan, J. (1989). "On the problem of penetration in particle methods". *Journal of Computational Physics*, 82(1):pp. 1 – 15. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/0021999189900326.

Monaghan, J. (1992). "Smoothed Particle Hydrodynamics". *Annual Review of Astronomy and astrophysics*, 30:pp. 543–574.

Monaghan, J. (1994). "Simulating Free Surface Flows with SPH". *J. Comp. Phys.*, 110(2):pp. 39–406.

Monaghan, J. (1997). "SPH and Riemann Solvers". *J. Comp. Phys.*, 136:pp. 298–307.

Monaghan, J. (2000). "SPH without a Tensile Instability". *Journal of Computational Physics*, 159(2):pp. 290 – 311. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/S0021999100964398.

Monaghan, J. (2005a). "Smoothed Particle Hydrodynamic simulations of shear flow". *Monthly Notices of the Royal Astronomical Society*, 365:pp. 199–213. URL http://dx.doi.org/10.1111/j.1365-2966.2005.09783.x.

Monaghan, J. (2005b). "Smoothed particle hydrodynamics". *Reports on Progress in Physics*, 68(8):pp. 1703–1759. URL http://stacks.iop.org/0034-4885/68/i=8/a=R01.

Monaghan, J. (2012). "Smoothed Particle Hydrodynamics and Its Diverse Applications". *Annual Review of Fluid Mechanics*, 44(1):pp. 323–346.

Monaghan, J., Cas, R., Kos, A., and Hallworth, M. (1999). "Gravity currents descending a ramp in a stratified tank". *Journal of Fluid Mechanics*, 139:p. 39.

Monaghan, J. and Gingold, R. A. (1983). "Shock Simulation by the particle method SPH". *Journal of Computational Physics*, 52(2):pp. 374–389.

Morris, J. (2000). “Simulating surface tension with smoothed particle hydrodynamics”. *International Journal for Numerical Methods in Fluids*, 33(3):pp. 333–353.

Morris, J. P. (1996). “A study of the stability properties of Smooth Particle Hydrodynamics”. *Publications of the Astronomical Society of Australia*, 13(1):p. 97?102.

Morris, J. P., Fox, P. J., and Zhu, Y. (1997). “Modeling Low Reynolds Number Incompressible Flows Using SPH”. *Journal of Computational Physics*, 136:pp. 214–226.

Moussa, B. and Villa, J. (2000). “Convergence of SPH method for scalar nonlinear conservation laws”. *Siam J. Numerical Analysis*.

Napoli, E., Marchis, M. D., Gianguzzi, C., Milici, B., and Monteleone, A. (2016). “A coupled Finite Volume-Smoothed Particle Hydrodynamics method for incompressible flows”. *Computer Methods in Applied Mechanics and Engineering*, 310:pp. 674–693. ISSN 0045-7825. URL `http://www.sciencedirect.com/science/article/pii/S0045782516308131`.

Nugent, S. and Posch, H. (2000). “Liquid drops and surface tension with smoothed particle applied mechanics”. *Physical Review E*, 62(4).

Okabe, A. (2016). “Spatial tessellations”. In *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, pp. 1–11. John Wiley and Sons, Ltd. ISBN 9781118786352.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, Ltd. ISBN 978-0-471-98635-5.

Oñate, E., Idelsohn, S. R., Celigueta, M. A., and Rossi, R. (2008). “Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows”. *Computer Methods in Applied Mechanics and Engineering*, 197(19):pp. 1777 – 1800. ISSN 0045-7825. Computational Methods in Fluid–Structure Interaction, URL `http://www.sciencedirect.com/science/article/pii/S0045782507002368`.

Oñate, E. and Owen, R. (2011). *Particle-Based Methods*, vol. 25 of *Computational Methods in Applied Sciences*. Springer Netherlands, 1 ed. ISBN 978-94-007-0735-1.

Ott, F. and Schnetter, E. (2003). “A modified SPH approach for fluids with large density differences”. *arXiv:physics/0303112*.

Quinlan, N. J., Lastiwka, M., and Basa, M. (2006). “Truncation error in mesh-free particle methods”. *International Journal for Numerical Methods in Engineering*, 66(13):pp. 2064–2085. URL `http://dx.doi.org/10.1002/nme.1617`.

Rapaport, D. C. (1996). “The art of molecular dynamics simulation”. *Computers in Physics*, 10(5):pp. 456–456. URL `https://doi.org/10.1063/1.4822471`.

Rycroft, C. (2009). “Voro++: A three-dimensional Voronoi cell library in C++”. Tech. rep., Lawerence Berkeley National Lab. Report Number: LBNL-1432E.

Serrano, M. (2002). *Modelos para la hidrodinámica mesoscópica de fluidos simples y complejos*. Ph.D. thesis, Universidad Nacional de Educación a Distancia.

Serrano, M. (2006). "Comparison between smoothed dissipative particle dynamics and Voronoi fluid particle model in a shear stationary flow". *Physica A: Statistical Mechanics and its Applications*, 362(1):pp. 204–209. ISSN 0378-4371. URL `http://www.sciencedirect.com/science/article/pii/S037843710500957X`.

Serrano, M. and Español, P. (2001). "Thermodynamically consistent mesoscopic fluid particle model". *Phys. Rev. E*, 64:p. 046115. URL `http://link.aps.org/doi/10.1103/PhysRevE.64.046115`.

Sethian, J. A. (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3. Cambridge university press.

Shapiro, P. R., Martel, H., Villumsen, J. V., and Kang, H. (1993). "Smoothed Particle Hydrodynamics and the Simulation of Galaxy and Large-Scale Structure Formation". *Revista Mexicana de Astronomia y Astrofisica*, 27.

Shapiro, P. R., Martel, H., Villumsen, J. V., and Owen, J. M. (1996). "Adaptive Smoothed Particle Hydrodynamics, with Application to Cosmology: Methodology". *Astrophysical Journal Supplement*, 103:p. 269.

Shobeyri, G. and Ardakani, R. R. (2017). "Improving accuracy of SPH method using Voronoi diagram". *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 41(3):pp. 345–350. ISSN 2364-1843.

Souto-Iglesias, A., Delorme, L., Pérez-Rojas, L., and Abril-Pérez, S. (2006). "Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics". *Ocean Engineering*, 33(11-12):pp. 1462–1484.

Souto-Iglesias, A., Maciá, F., González, L. M., and Cercos-Pita, J. L. (2013). "On the consistency of MPS". *Computer Physics Communications*, 184(3):pp. 732 – 745. ISSN 0010-4655. URL `http://www.sciencedirect.com/science/article/pii/S0010465512003852`.

Springel, V. (2010a). "E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh". *Monthly Notices of the Royal Astronomical Society*, 401(2):p. 791.

Springel, V. (2010b). "Smoothed Particle Hydrodynamics in Astrophysics". *Annual Review of Astronomy and Astrophysics*, 48(1):pp. 391–430. URL `https://doi.org/10.1146/annurev-astro-081309-130914`.

Sulsky, D., Chen, Z., and Schreyer, H. L. (1994). "A particle method for history-dependent materials". *Computer Methods in Applied Mechanics and Engineering*, 118(1):pp. 179–196. ISSN 0045-7825. URL `http://www.sciencedirect.com/science/article/pii/0045782594901120`.

Sun, P., Colagrossi, A., Marrone, S., and Zhang, A. (2017). "The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme". *Computer Methods in Applied Mechanics and Engineering*, 315(Supplement C):pp. 25 – 49. ISSN 0045-7825. URL http://www.sciencedirect.com/science/article/pii/S0045782516309112.

Swegle, J. W., Hicks, D. L., and Attaway, S. W. (1995). "Smoothed Particle Hydrodynamics Stability Analysis". *Journal of Computational Physics*, 116:pp. 123–134.

Takeda, H., Miyama, S. M., and Sekiya, M. (1994). "Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics". *Progress of Theoretical Physics*, 92(5):pp. 939–960.

Vacondio, R., Rogers, B., Stansby, P., Mignosa, P., and Feldman, J. (2013). "Variable resolution for SPH: A dynamic particle coalescing and splitting scheme". *Computer Methods in Applied Mechanics and Engineering*, 256(Supplement C):pp. 132 – 148. ISSN 0045-7825. URL http://www.sciencedirect.com/science/article/pii/S0045782512003842.

Violeau, D. (2009). "Dissipative forces for Lagrangian models in computational fluid dynamics and application to smoothed-particle hydrodynamics". *Phys. Rev. E*, 80:p. 036705. URL https://link.aps.org/doi/10.1103/PhysRevE.80.036705.

Violeau, D. (2012). *Fluid mechanics and the SPH method: theory and applications*. Oxford University Press.

Voronoi, G. (1908). "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs". *Journal für die reine und angewandte Mathematik*, 134:pp. 198–287. ISSN 0075-4102.

Wagner, G. J. and Liu, W. K. (2000). "Turbulence simulation and multiple scale subgrid models". *Computational Mechanics*, 25(2):pp. 117–136. ISSN 1432-0924.

Wang, L. (2016). *High-Performance Discrete-Vortex Algorithms for Unsteady Viscous-Fluid Flows near Moving Boundaries*. Ph.D. thesis, University of California at Berkeley.

Wang, L. and Yeung, R. W. (2016a). "Investigation of full and partial ground effects on a flapping foil hovering above a finite-sized platform". *Physics of Fluids*, 28(7):p. 071902. URL https://aip.scitation.org/doi/abs/10.1063/1.4954656.

Wang, L. and Yeung, R. W. (2016b). "On the performance of a micro-scale Bach-type turbine as predicted by discrete-vortex simulations". *Applied Energy*, 183:pp. 823 – 836. ISSN 0306-2619. URL http://www.sciencedirect.com/science/article/pii/S0306261916312946.

Watson, D. F. (1981). "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes". *The Computer Journal*, 24(2):pp. 167–172. /oup/backfile/content_public/journal/comjnl/24/2/10.1093/comjnl/24.2.167/2/240167.pdf.

Welton, W. C. (1998). "Two-Dimensional PDF/SPH simulations of compressible turbulent flows". *Journal of Computational Physics*, 139(2):pp. 410 – 443. ISSN 0021-9991. URL http://www.sciencedirect.com/science/article/pii/S0021999197958782.

Yeung, R. W. (1982). “Numerical methods in free-surface flows”. *Annual Review of Fluid Mechanics*, 14(1):pp. 395–442.

Yildiz, M., Rook, R. A., and Suleman, A. (2009). “SPH with the multiple boundary tangent method”. *International Journal for Numerical Methods in Engineering*, 77(10):pp. 1416–1438. ISSN 1097-0207.

Zhou, Z., De Kat, J., and Buchner, B. (1999). “A nonlinear 3-D approach to simulate green water dynamics on deck”. In *Proc. 7th International Symposium on Numerical Ship Hydrodynamics, Report*, vol. 7.

Appendix A

Artificial *vs.* Real viscosity

A.1 General

Following the derivation from Español and Revenga (2003) and Violeau (2009), we can relate the artificial viscosity (AV) formulation from Eq. (3.14) to the viscous term in the Navier-Stokes momentum Eq. (2.9). In addition, this relation allows us to determine an alternative way to express the Laplacian operator to directly follow the smoothing procedure, which is more consistent with its physical meaning as discussed by Monaghan (2005b).

Let's start with the expression of the viscous term used, as defined in Sec. 3.1.1:

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV} = -\sum_b m_b \frac{\alpha\, \overline{h}_{ab}\, \overline{c}_{ab}\, \overline{\rho}_{ab}}{\rho_a\, \rho_b} \frac{\boldsymbol{u}_{ab} \cdot \boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2} \nabla_a W_{ab} \,. \tag{A.1}$$

Turning back the summation into an integral, and taking h, c, and ρ as constants, we can switch to the continuum field

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV} = -\int_\Omega \alpha\, h\, c\, \frac{(\boldsymbol{u}-\boldsymbol{u}^*) \cdot (\boldsymbol{r}-\boldsymbol{r}^*)}{\|(\boldsymbol{r}-\boldsymbol{r}^*)\|^2} \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*, h)\, \mathrm{d}\mathcal{V}^* \,, \tag{A.2}$$

where

$$\begin{aligned} m_b \frac{\overline{\rho}_{ab}}{\rho_a\, \rho_b} &\rightarrow \frac{m}{\rho} = \mathcal{V} \rightarrow \mathrm{d}\mathcal{V}^* \,, \\ \boldsymbol{u}_{ab} &\rightarrow (\boldsymbol{u}-\boldsymbol{u}^*) \,, \\ \boldsymbol{r}_{ab} &\rightarrow (\boldsymbol{r}-\boldsymbol{r}^*) \,. \end{aligned}$$

Looking at the kernel gradient, we saw in Sec. 3.1.2 how we can use the following relations

$$\begin{aligned} W(\boldsymbol{r}-\boldsymbol{r}^*, h) &= \frac{k_k}{h^D} M(q) \,, \\ \nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*, h) &= (\boldsymbol{r}-\boldsymbol{r}^*) \frac{k_k}{q\, h^{D+2}} \frac{\mathrm{d}M(q)}{\mathrm{d}q} = (\boldsymbol{r}-\boldsymbol{r}^*) F(q) \,, \end{aligned}$$

where D is the number of spatial dimensions, $q = \frac{\|\boldsymbol{r}-\boldsymbol{r}^*\|}{h}$, and k_k is a normalizing coefficient that depends on D and the kernel used. Using index notation as in Español and Revenga (2003), where the super-indexes indicate the tensor component, this integral can be rewritten as

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\, i} = -\alpha\, h\, c \int_\Omega \frac{(\boldsymbol{u}-\boldsymbol{u}^*)^j\, (\boldsymbol{r}-\boldsymbol{r}^*)^j}{(h\, q)^2} (\boldsymbol{r}-\boldsymbol{r}^*)^i F(q) \mathrm{d}\mathcal{V}^* \,. \tag{A.3}$$

By Taylor-expanding $\boldsymbol{u}^*$ around $\boldsymbol{r}$, we obtain

$$(\boldsymbol{u}^*)^j = (\boldsymbol{u})^j + \nabla^k \boldsymbol{u}^j (\boldsymbol{r}-\boldsymbol{r}^*)^k + \frac{1}{2} \nabla^k \nabla^l \boldsymbol{u}^j (\boldsymbol{r}-\boldsymbol{r}^*)^k (\boldsymbol{r}-\boldsymbol{r}^*)^l + O\left(\nabla^3 \boldsymbol{u}^j : (\boldsymbol{r}-\boldsymbol{r}^*)^3\right) \,, \tag{A.4}$$

where

$$\nabla^k = \frac{\partial}{\partial x_k}\,,$$

being x_k the k-th coordinate in the reference system used. The higher order terms are negligible for sufficiently smooth fields, and using this expansion in Eq. (A.3), we arrive at

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\,i} \approx -\alpha\, h\, c \int_\Omega \frac{\left[\nabla^k \boldsymbol{u}^j (\boldsymbol{r}-\boldsymbol{r}^*)^k + \frac{1}{2}\nabla^k\nabla^l \boldsymbol{u}^j (\boldsymbol{r}-\boldsymbol{r}^*)^k (\boldsymbol{r}-\boldsymbol{r}^*)^l\right] (\boldsymbol{r}-\boldsymbol{r}^*)^j}{(h\,q)^2} (\boldsymbol{r}-\boldsymbol{r}^*)^i F(q) \mathrm{d}\mathcal{V}^*\,. \tag{A.5}$$

We can express this integral in a more compact way by defining $(\boldsymbol{r}-\boldsymbol{r}^*) = h\,\boldsymbol{q}$ and $\mathrm{d}\mathcal{V}^* = h^D\,\mathrm{d}Q$, with D being the number of dimension of the domain space, so

$$\begin{aligned}\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\,i} &\approx -\alpha\, h\, c \int_\Omega \frac{\left[\nabla^k \boldsymbol{u}^j\, h\, \boldsymbol{q}^k + \frac{1}{2}\nabla^k\nabla^l \boldsymbol{u}^j\, h^2\, \boldsymbol{q}^k \boldsymbol{q}^l\right] h\, \boldsymbol{q}^j}{(h\,q)^2}\, h\, \boldsymbol{q}^i F(q)\, h^D\, \mathrm{d}Q \\ &\approx -\alpha\, h\, c \left(h^{D+1}\, \nabla^k \boldsymbol{u}^j \int_\Omega \frac{\left[\boldsymbol{q}^i\, \boldsymbol{q}^j\, \boldsymbol{q}^k\right]}{q^2} F(q)\, \mathrm{d}Q + \frac{1}{2} h^{D+2} \nabla^k\nabla^l \boldsymbol{u}^j \int_\Omega \frac{\left[\boldsymbol{q}^i\, \boldsymbol{q}^j\, \boldsymbol{q}^k\, \boldsymbol{q}^l\right]}{q^2} F(q)\, \mathrm{d}Q\right).\end{aligned} \tag{A.6}$$

It can be shown how the first integral in Eq. (A.6) vanishes by isotropy of the third order tensor, reducing it to

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\,i} \approx -\frac{1}{2}\alpha\, h\, c \left(\nabla^k\nabla^l \boldsymbol{u}^j \int_\Omega h^{D+2} \frac{\left[\boldsymbol{q}^i\, \boldsymbol{q}^j\, \boldsymbol{q}^k\, \boldsymbol{q}^l\right]}{q^2} F(q)\, \mathrm{d}Q\right). \tag{A.7}$$

This fourth order tensor integral can be directly computed using the Kernel formulation. Again, we can show how the only components of the fourth order tensor that render non-zero integral outputs are of the type $[\boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha]$ and $\left[\boldsymbol{q}^\beta\, \boldsymbol{q}^\beta\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\right]$, with α and β going through all the components (see Sec. A.2).

$$-\int_\Omega h^{D+2} \frac{[\boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha]}{q^2} F(q)\, \mathrm{d}Q \;=\; k_\alpha = 3k_\beta\,, \tag{A.8}$$

$$-\int_\Omega h^{D+2} \frac{\left[\boldsymbol{q}^\beta\, \boldsymbol{q}^\beta\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\right]}{q^2} F(q)\, \mathrm{d}Q \;=\; k_\beta\,. \tag{A.9}$$

We can therefore contract the RHS of Eq. (A.7), which becomes simply

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\,i} \approx \frac{1}{2}\alpha\, h\, c \sum_{j=1}^{D} \left(3k_\beta \delta_{ij} \nabla^i\nabla^i \boldsymbol{u}^j + k_\beta (1-\delta_{ij}) \left(\nabla^i\nabla^j \boldsymbol{u}^j + \nabla^j\nabla^i \boldsymbol{u}^j + \nabla^j\nabla^j \boldsymbol{u}^i\right)\right). \tag{A.10}$$

Noting that

$$\begin{gathered}\nabla^j\nabla^i = \nabla^i\nabla^j\,,\\ \delta_{ij}\nabla^i\nabla^i v^j = \delta_{ij}\nabla^i\nabla^j v^j = \delta_{ij}\nabla^j\nabla^j v^i = \nabla^i\nabla^i v^i\,,\end{gathered}$$

we easily reach

$$\begin{aligned}\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV,\,i} &\approx \frac{1}{2}\alpha\, h\, c \sum_{j=1}^{D}\left(k_\beta \nabla^j\nabla^j\boldsymbol{u}^i + 2k_\beta\nabla^i\nabla^j\boldsymbol{u}^j + (k_\alpha - 3k_\beta)\nabla^i\nabla^i\boldsymbol{u}^j\delta_{ij}\right)\\ &\approx \frac{1}{2}\alpha\, h\, c \sum_{j=1}^{D}\left(k_\beta \nabla^j\nabla^j\boldsymbol{u}^i + 2k_\beta\nabla^i\nabla^j\boldsymbol{u}^j\right),\end{aligned} \tag{A.11}$$

since $k_\alpha = 3k_\beta$ (Sec. A.2). Furthermore, Sec. A.3 shows how $k_\beta = \frac{1}{D+2}$ for all kernels. Recovering the vector notation, it becomes directly

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV} \approx \frac{1}{2(D+2)}\,\alpha\, h\, c\left[\nabla^2\boldsymbol{u} + 2\nabla(\nabla\cdot\boldsymbol{u})\right], \tag{A.12}$$

which matches perfectly the real viscous terms in the Navier-Stokes equations,

$$\rho\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{VI} = \mu\nabla^2\boldsymbol{u} + (\mu+\lambda)\nabla(\nabla\cdot\boldsymbol{u}), \tag{A.13}$$

with

$$\mu = \lambda = \frac{1}{2\,(D+2)}\,\frac{\alpha\, h\, c}{\rho}. \tag{A.14}$$

We can extract the following conclusions:

- The formulation used for artificial viscosity effectively includes both shear and compressibility in a single term, beneficial to fulfill its goal of dampening the different sources of numerical noise.

- If it is assumed that the fluid properties satisfy $\mu = \lambda$, it can be used to model very closely the viscous diffusion in the momentum equation.

- Even if another relation is assumed where $\mu \neq \lambda$, like the commonly followed Stokes hypothesis of zero bulk viscosity ($\mu_B = \frac{2}{3}\mu + \lambda = 0$), the compressiblity component is usually negligible in weakly compressible fluid, as discussed by Marrone et al. (2013) or Colagrossi et al. (2013)). Therefore, we can still use as a good approximation

  $$\frac{1}{2(D+2)}\,\alpha\, h\, c\left[\nabla^2\boldsymbol{u} + 2\nabla(\nabla\cdot\boldsymbol{u})\right] \approx \frac{1}{2(D+2)}\,\alpha\, h\, c\,\nabla^2\boldsymbol{u}, \tag{A.15}$$

  which equaled to a slightly simplified artificial viscous formulation provides a good approximation of the Laplacian operator in its SPH form:

  $$\frac{1}{2(D+2)}\,\alpha\, h\, c\,\nabla^2\boldsymbol{u}_a \approx -\sum_b m_b \frac{\alpha\, h\, c\,\overline{\rho}_{ab}}{\rho_a\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\nabla_a W_{ab}, \tag{A.16}$$

  $$\nabla^2\boldsymbol{u}_a \approx -\sum_b 2(D+2)\, m_b \frac{\overline{\rho}_{ab}}{\rho_a\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\nabla_a W_{ab}. \tag{A.17}$$

A.2 $k_\alpha = 3k_\beta$ PROOF

Our goal is determine the values of the fourth order tensor resulting from the integrals of the type:

$$-\int_\Omega h^{D+2} \frac{\left[\boldsymbol{q}^i\,\boldsymbol{q}^j\,\boldsymbol{q}^k\,\boldsymbol{q}^l\right]}{q^2} F(q)\,\mathrm{d}Q\,. \tag{A.18}$$

For a normalized D-dimension space, we can define a generalized D-spherical coordinate system, $\{q, \theta_1, \ldots, \theta_{D-1}\}$, with $q \in [0, \infty)$, $\theta_1 \in [0, 2pi)$, and $\theta_m \in [0, \pi]$ for $m > 1$, which can be converted to Cartesian as[1]

$$\begin{cases} x_1 = qx_1^* = q\cos\theta_1\,, \\ x_2 = qx_2^* = q\sin\theta_1\cos\theta_2\,, \\ \vdots \\ x_i = qx_i^* = q\prod\limits_{m=1}^{i-1}\sin\theta_m\cos\theta_i\,, \\ \vdots \\ x_{D-1} = qx_{D-1}^* = q\prod\limits_{m=1}^{D-2}\sin\theta_m\cos\theta_{D-1}\,, \\ x_D = qx_D^* = q\prod\limits_{m=1}^{D-1}\sin\theta_m\,. \end{cases} \tag{A.19}$$

and whose differential volume element is given by

$$\mathrm{d}Q = q^{D-1}\sin^{D-2}\theta_1\sin^{D-3}\theta_2\,\cdots\,\sin\theta_{D-2}\,\mathrm{d}r\,\mathrm{d}\theta_1\cdots\mathrm{d}\theta_{D-1} = q^{D-1}\mathrm{d}q\prod_{m=1}^{D-2}\sin^{D-1-m}\theta_m\,\mathrm{d}\theta_m\,\mathrm{d}\theta_{D-1}\,. \tag{A.20}$$

Under this coordinate system, we can rewrite the initial integral in Eq. (A.18) as

$$\left[\boldsymbol{q}^i\,\boldsymbol{q}^j\,\boldsymbol{q}^k\,\boldsymbol{q}^l\right] = q^4\left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*l}\right]\,, \tag{A.21}$$

$$\int_Q h^{D+2}\,q^{D+1}\,F(q)\mathrm{d}q \int\limits_{\substack{\theta_1=0\\ \theta_m=0\\ \theta_{D-1}=0}}^{\substack{\theta_1=2\pi\\ \theta_m=\pi\\ \theta_{D-1}=\pi}} \left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*l}\right]\prod_{m=1}^{D-2}\sin^{D-1-m}\theta_m\,\mathrm{d}\theta_m\,\mathrm{d}\theta_{D-1}\,, \tag{A.22}$$

where Q represents the kernel support radius in the normalized space. The integration over q is the same no matter which combination of i, j, k, and l. We will refer to it as

$$K_q = \int_Q h^{D+2}\,q^{D+1}\,F(q)\mathrm{d}q\,. \tag{A.23}$$

---

[1] `https://en.wikipedia.org/wiki/N-sphere`

Therefore, all the differences should come from the integration over the angular domains. We can expand the integrand as

$$\left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*l}\right] = \left(\prod_{m=1}^{i-1}\sin\theta_m\cos\theta_i\right)\left(\prod_{m=1}^{j-1}\sin\theta_m\cos\theta_j\right)\left(\prod_{m=1}^{k-1}\sin\theta_m\cos\theta_k\right)\left(\prod_{m=1}^{l-1}\sin\theta_m\cos\theta_l\right). \tag{A.24}$$

Without any loss of generality, we can order the indexes such that always $i \leq j \leq k \leq l$. Let's analyze the following possible scenarios, where we use α, β, γ, and δ to refer to distinct values but in no particular order:

At least one index unique. This situation happens when

- $[\boldsymbol{q}^\delta, \boldsymbol{q}^\gamma, \boldsymbol{q}^\beta, \boldsymbol{q}^\alpha]$: All indexes are different, as shown in Eq. (A.25).
- $[\boldsymbol{q}^\gamma, \boldsymbol{q}^\gamma, \boldsymbol{q}^\beta, \boldsymbol{q}^\alpha]$: Only two are equal. Eq. (A.26) shows one possible combination, with the only change when $i < j = k < l$ or $i = j < k < l$, consisting on where the $\cos^2\theta$ is placed.
- $[\boldsymbol{q}^\beta, \boldsymbol{q}^\beta, \boldsymbol{q}^\beta, \boldsymbol{q}^\alpha]$: Three indexes are equal. Eq. (A.27) shows one possible combination, with the only change being where $\cos^3\theta$ is placed when $i = j = k < l$.

$$i<j<k<l \;:\; \left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*l}\right] = \left(\prod_{m=1}^{i-1}\sin^4\theta_m\right)\sin^3\theta_i\cos\theta_i\left(\prod_{m=i+1}^{j-1}\sin^3\theta_m\right)\sin^2\theta_j\cos\theta_j \left(\prod_{m=j+1}^{k-1}\sin^2\theta_m\right)\sin\theta_k\cos\theta_k\left(\prod_{m=k+1}^{l-1}\sin\theta_m\right)\cos\theta_l\,, \tag{A.25}$$

$$i<j<k=l \;:\; \left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*k}\right] = \left(\prod_{m=1}^{i-1}\sin^4\theta_m\right)\sin^3\theta_i\cos\theta_i \left(\prod_{m=i+1}^{j-1}\sin^3\theta_m\right)\sin^2\theta_j\cos\theta_j\left(\prod_{m=j+1}^{k-1}\sin^2\theta_m\right)\cos^2\theta_k\,, \tag{A.26}$$

$$i<j=k=l \;:\; \left[x^{*i}\,x^{*j}\,x^{*j}\,x^{*j}\right] = \left(\prod_{m=1}^{i-1}\sin^4\theta_m\right)\sin^3\theta_i\cos\theta_i\left(\prod_{m=i+1}^{j-1}\sin^3\theta_m\right)\cos^3\theta_j\,. \tag{A.27}$$

Equations (A.25), (A.26), and (A.27) are given for $l < D$. However, the only difference when $l = D$ is in the last term, which becomes $\sin^n\theta$ instead of $\cos^n\theta$, where n refers here to a general integer value to include all cases.

We should note how the common characteristic of all expressions is that at least display one coefficient of the type $\sin^n\theta\cos\theta$. Since no $\cos\theta$ exists in $\mathrm{d}\boldsymbol{r}$, these coefficients will directly lead to integrals of the type

$$\int_0^{\pi,2\pi}\sin^n\theta\cos\theta\mathrm{d}\theta = \left.\frac{\sin^{n+1}\theta}{n+1}\right|_0^{\pi,2\pi} = 0 \quad \forall\, n\,. \tag{A.28}$$

We conclude therefore that all this type of combinations will lead to zero-integrals.

Indexes equal by pairs: $[\boldsymbol{q}^\beta, \boldsymbol{q}^\beta, \boldsymbol{q}^\alpha, \boldsymbol{q}^\alpha]$, as shown in Eq. (A.29).

$$i = j < k = l \ :\ \left[x^{*i}\,x^{*i}\,x^{*k}\,x^{*k}\right] = \left(\prod_{m=1}^{i-1} \sin^4\theta_m\right) \sin^2\theta_i \cos^2\theta_i \left(\prod_{m=i+1}^{k-1} \sin^2\theta_m\right) \cos^2\theta_k\,. \tag{A.29}$$

None of the previous situation occurs, which lead to non-zero integral. Eq. (A.29) is given for $k \neq D$. If $k = D$, the only change is that the last term becomes $\sin^2\theta_{D-1}$ instead of $\cos^2\theta_{D-1}$. Substituting them in the angular integrals from Eq. (A.24), we can use the following algebraic properties[2] for the different integral patterns:

$$m \in [1, (i-1)] \ :\ \int_0^{\pi,2\pi} \sin^{n+4}\theta\,\mathrm{d}\theta_m = \frac{(n+3)(n+1)}{(n+4)(n+2)} \int_0^{\pi,2\pi} \sin^n\theta_m\,\mathrm{d}\theta_m\,, \tag{A.30}$$

$$m = i \ :\ \int_0^{\pi,2\pi} \sin^{n+2}\theta_m\,\cos^2\theta_m\,\mathrm{d}\theta_m = \frac{(n+1)}{(n+4)(n+2)} \int_0^{\pi,2\pi} \sin^n\theta_m\,\mathrm{d}\theta_m\,, \tag{A.31}$$

$$m \in [(i+1), (k-1)] \ :\ \int_0^{\pi} \sin^{n+2}\theta\,\mathrm{d}\theta_m = \frac{(n+1)}{(n+2)} \int_0^{\pi} \sin^n\theta_m\,\mathrm{d}\theta_m\,, \tag{A.32}$$

$$m = k \neq D \ :\ \int_0^{\pi} \sin^n\theta_m\,\cos^2\theta_m \mathrm{d}\theta_m = \frac{1}{(n+2)} \int_0^{\pi} \sin^n\theta_m\,\mathrm{d}\theta_m\,, \tag{A.33}$$

$$m = k = D \ :\ \int_0^{\pi} \sin^2\theta_{D-1}\mathrm{d}\theta_m = \frac{\pi}{2} = \frac{1}{(n+2)} \int_0^{\pi} \sin^n\theta_{D-1}\mathrm{d}\theta_{D-1}\ (n=0)\,, \tag{A.34}$$

so we take the coefficients due to x^* outside the integrals and arrive at

$$\int_{\substack{\theta_1=0\\ \theta_m=0\\ \theta_{D-1}=0}}^{\substack{\theta_1=2\pi\\ \theta_m=\pi\\ \theta_{D-1}=\pi}} \left[x^{*i}\,x^{*j}\,x^{*k}\,x^{*l}\right] \prod_{m=1}^{D-2} \sin^{D-1-m}\theta_m\,\mathrm{d}\theta_m\,\mathrm{d}\theta_{D-1} = K_{\beta\beta\alpha\alpha}\,K_\theta\,, \tag{A.35}$$

$$K_{\beta\beta\alpha\alpha} = \left(\prod_{m=1}^{i-1} \frac{(D-m+2)(D-m)}{(D-m+3)(D-m+1)}\right) \frac{(D-i)}{(D-i+3)(D-i+1)} \left(\prod_{m=i+1}^{k-1} \frac{(D-m)}{(D-m+1)}\right) \frac{1}{(D-k+1)}\,, \tag{A.36}$$

$$K_\theta = \int_{\substack{\theta_1=0\\ \theta_m=0}}^{\substack{\theta_1=2\pi\\ \theta_m=\pi}} \prod_{m=1}^{D-2} \sin^{D-1-m}\theta_m\,\mathrm{d}\theta_m \int_0^{\pi} \mathrm{d}\theta_{D-1} = \frac{2\pi^{D/2}}{\Gamma\left(\frac{D}{2}\right)}\,, \tag{A.37}$$

where we used the definition of the surface area of an D-dimensional ball for K_θ, being Γ the gamma function[3]. Furthermore, we can simplify $K_{\beta\beta\alpha\alpha}$ by noticing that we can rewrite it as

$$K_{\beta\beta\alpha\alpha} = \frac{\prod_{m=1}^{i-1}(D-m+2)}{\prod_{m=1}^{i}(D-m+3)}\,\frac{\prod_{m=1}^{k-1}(D-m)}{\prod_{m=1}^{k}(D-m+1)}\,, \tag{A.38}$$

---

[2] The can be easily obtained integrating by parts

[3] $\Gamma(1/2) = \sqrt{\pi}$; $\Gamma(1) = 1$; $\Gamma(x+1) = x\Gamma(x)$

and since

$$\prod_{m=1}^{i}(D-m+3) = \prod_{m=0}^{i-1}(D-m+2) = (D+2)\prod_{m=1}^{i-1}(D-m+2)\,, \tag{A.39}$$

$$\prod_{m=1}^{k}(D-m+1) = \prod_{m=0}^{k-1}(D-m) = D\prod_{m=1}^{k-1}(D-m)\,, \tag{A.40}$$

it simply reduces to

$$K_{\beta\beta\alpha\alpha} = \frac{1}{D\,(D+2)}\,, \tag{A.41}$$

and based on Eq. (A.9) we arrive at

$$k_\beta = -K_q\, K_{\beta\beta\alpha\alpha}\, K_\theta = -\frac{K_q\, K_\theta}{D\,(D+2)}\,. \tag{A.42}$$

All indexes equal: $[\boldsymbol{q}^\alpha, \boldsymbol{q}^\alpha, \boldsymbol{q}^\alpha, \boldsymbol{q}^\alpha]$, as shown in Eq. (A.43).

$$i = j = k = l \;:\; \left[x^{*i}\, x^{*i}\, x^{*k}\, x^{*k}\right] = \left(\prod_{m=1}^{i-1} \sin^4\theta_m\right)\cos^4\theta_i\,. \tag{A.43}$$

Doing similar analysis as in the previous case, we use now the following algebraic properties in addition to Eq. (A.30) for $m \in [1, (i-1)]$:

$$m = i \neq D \;:\; \int_0^\pi \sin^n\theta_m\, \cos^4\theta_m \mathrm{d}\theta_m = \frac{3}{(n+4)(n+2)}\int_0^\pi \sin^n\theta_m\, \mathrm{d}\theta_m\,, \tag{A.44}$$

$$m = i = D \;:\; \int_0^\pi \sin^4\theta_{D-1}\mathrm{d}\theta_m = \frac{3\pi}{8} = \frac{3}{(n+4)(n+2)}\int_0^\pi \sin^n\theta_{D-1}\mathrm{d}\theta_{D-1}\;(n=0)\,. \tag{A.45}$$

Substituting these expression in the angular integral renders now

$$K_{\alpha\alpha\alpha\alpha} = \left(\prod_{m=1}^{i-1}\frac{(D-m+2)(D-m)}{(D-m+3)(D-m+1)}\right)\frac{3}{(D-i+3)(D-i+1)} = \frac{3\,\prod_{m=1}^{i-1}(D-m+2)(D-m)}{\prod_{m=1}^{i}(D-m+3)(D-m+1)}\,, \tag{A.46}$$

which using the same properties than before reduces to

$$K_{\alpha\alpha\alpha\alpha} = \frac{3}{D\,(D+2)}\,, \tag{A.47}$$

and from Eq. (A.8):

$$k_\alpha = -K_q\, K_{\alpha\alpha\alpha\alpha}\, K_\theta = -\frac{3K_q\, K_\theta}{D\,(D+2)} = 3k_\beta\,, \tag{A.48}$$

as we wanted to demonstrate.

A.3 k_k AND k_β

The results from the previous section[4] show that we can express k_β as

$$k_\beta = -\int_Q h^{D+2} \frac{\left[\boldsymbol{q}^\beta\, \boldsymbol{q}^\beta\, \boldsymbol{q}^\alpha\, \boldsymbol{q}^\alpha\right]}{q^2}\, F(q)\, \mathrm{d}Q = -\frac{K_q\, K_\theta}{D\,(D+2)}\,, \tag{A.49}$$

with

$$K_q = \int_Q h^{D+2}\, q^{D+1}\, F(q) \mathrm{d}q\,,$$

$$K_\theta = \int\limits_{\substack{\theta_1=0\\ \theta_m=0}}^{\substack{\theta_1=2\pi\\ \theta_m=\pi}} \prod_{m=1}^{D-2} \sin^{D-1-m}\theta_m\, \mathrm{d}\theta_m \int_0^\pi \mathrm{d}\theta_{D-1} = \frac{2\pi^{D/2}}{\Gamma\left(\frac{D}{2}\right)}\,.$$

The only remaining step to determine k_β is calculate K_q. At first sight, it depends on $F(q)$, but if we use its definition as a function of the kernel function from Eq. (3.21),

$$F(q) = \frac{k_k}{h^{D+2}\, q} \frac{\mathrm{d}M(q)}{\mathrm{d}q}\,, \tag{A.50}$$

we can rewrite K_q using integration by parts as

$$K_q = \int_Q h^{D+2}\, q^{D+1}\, \frac{k_k}{h^{D+2}\, q} \frac{\mathrm{d}M(q)}{\mathrm{d}q}\, \mathrm{d}q\,, = k_k \int_Q q \frac{\mathrm{d}M(q)}{\mathrm{d}q}\, \mathrm{d}q = -k_k \int_Q D\, q^{D-1} M(q)\, \mathrm{d}q\,, \tag{A.51}$$

since $M(q)$ is zero beyond its support radius. In addition, we should notice that k_k is a normalization coefficient used to satisfy $\int_\Omega W(\boldsymbol{r} - \boldsymbol{r}^*, h)\mathrm{d}\boldsymbol{r}^* = 1$, so expanding W as a function of M from Eq. (3.19), and normalizing $\mathrm{d}\boldsymbol{r}^*$ with h as before, we arrive at

$$\int_Q \frac{k_k}{h^D} M(q)\, h^D\, \mathrm{d}\boldsymbol{q} = k_k \int_Q M(q)\, q^{D-1} \mathrm{d}q\, \overbrace{\int\limits_{\substack{\theta_1=0\\ \theta_m=0}}^{\substack{\theta_1=2\pi\\ \theta_m=\pi}} \prod_{m=1}^{D-2} \sin^{D-1-m}\theta_m\, \mathrm{d}\theta_m \int_0^\pi \mathrm{d}\theta_{D-1}}^{K_\theta}\,, \tag{A.52}$$

$$k_k = \frac{1}{K_\theta \int_Q M(q)\, q^{D-1} \mathrm{d}q} = \frac{\Gamma\left(\frac{D}{2}\right)}{2\pi^{D/2} \int_Q M(q)\, q^{D-1} \mathrm{d}q}\,. \tag{A.53}$$

Substituting it in Eq. (A.51), and independently of the kernel function used, K_q and k_β simply reduce to

$$K_q = -\frac{D}{K_\theta}\,, \tag{A.54}$$

$$k_\beta = -\frac{-\frac{D}{K_\theta}\, K_\theta}{D\,(D+2)} = \frac{1}{D+2}\,. \tag{A.55}$$

---

[4] See Eqs. (A.9, A.23, A.37, A.41, and A.42)

APPENDIX B

VPH GRADIENT APPROXIMATION

Below is summarized the derivation presented by Hess and Springel (2010) that leads to the Voronoi gradient operator. It should be stressed here that the notation $\boldsymbol{e}_{ab}$, with the vector pointing from a to b usually used as notation for SPH, is the opposite than Hess and Springel (2010)'s notation. In their article, a vector subindex ab implies that is obtained by subtracting the value in particle b from that in particle a.

We start by proving Eq. (3.54),

$$\int_{\mathcal{V}_a} \nabla f(\boldsymbol{r}^*)\,\mathrm{d}\mathcal{V}^* = \int_{\mathcal{S}_a} \boldsymbol{r}^* \left(\nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^*\right) - \int_{\mathcal{V}_a} \boldsymbol{r}^*\, \nabla^2 f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* \,,$$

which is obtained by applying the divergence theorem on $(\mathbf{1} \cdot \boldsymbol{r}*)\, \nabla f(\boldsymbol{r}^*)$, being $f(\boldsymbol{r}^*)$ a scalar field and $\mathbf{1}$ a random constant unit vector, such that

$$\int_{\mathcal{V}_a} \nabla \cdot \left[(\mathbf{1} \cdot \boldsymbol{r}^*)\, \nabla f(\boldsymbol{r}^*)\right] \mathrm{d}\mathcal{V}^* = \int_{\mathcal{S}_a} \left[(\mathbf{1} \cdot \boldsymbol{r}^*)\, \nabla f(\boldsymbol{r}^*)\right] \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* \,. \tag{B.1}$$

We can expand the integrand in the LHS as

$$\nabla \cdot \left[(\mathbf{1} \cdot \boldsymbol{r}^*)\, \nabla f(\boldsymbol{r}^*)\right] = \nabla(\mathbf{1} \cdot \boldsymbol{r}^*) \cdot \nabla f(\boldsymbol{r}^*) + (\mathbf{1} \cdot \boldsymbol{r}^*)\nabla^2 f(\boldsymbol{r}^*) \,. \tag{B.2}$$

Noting that $\nabla \boldsymbol{r}^* = \mathbb{I}$ and $\nabla \mathbf{1} = \mathbb{O}$, being $\mathbb{I}$ and $\mathbb{O}$ the identity and zero tensors, respectively,

$$\nabla(\mathbf{1} \cdot \boldsymbol{r}^*) = \nabla \mathbf{1} \cdot \boldsymbol{r}^* + \mathbf{1} \cdot \nabla \boldsymbol{r}^* = \mathbf{1} \cdot \mathbb{I} = \mathbf{1} \,, \tag{B.3}$$

$$\nabla \cdot \left[(\mathbf{1} \cdot \boldsymbol{r}^*)\, \nabla f(\boldsymbol{r}^*)\right] = \mathbf{1} \cdot \nabla f(\boldsymbol{r}^*) + (\mathbf{1} \cdot \boldsymbol{r}^*)\nabla^2 f(\boldsymbol{r}^*) \,. \tag{B.4}$$

Using Eq. (B.4) in the left hand side of Eq. (B.1), we arrive at

$$\int_{\mathcal{V}_a} \mathbf{1} \cdot \nabla f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* + \int_{\mathcal{V}_a} (\mathbf{1} \cdot \boldsymbol{r}^*)\nabla^2 f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* = \int_{\mathcal{S}_a} \left[(\mathbf{1} \cdot \boldsymbol{r}^*)\, \nabla f(\boldsymbol{r}^*)\right] \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* \,. \tag{B.5}$$

$\mathbf{1}$ is constant, so it can be taken out of the integral on both sides. Rearranging terms,

$$\cancel{\mathbf{1}\cdot} \int_{\mathcal{V}_a} \nabla f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* = \cancel{\mathbf{1}\cdot} \int_{\mathcal{S}_a} \boldsymbol{r}^*\, \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* - \cancel{\mathbf{1}\cdot} \int_{\mathcal{V}_a} \boldsymbol{r}^*\, \nabla^2 f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* \,. \tag{B.6}$$

Basically, "$\mathbf{1}\cdot$" implies a projection of the vectors from the integrals into the direction given by $\mathbf{1}$. Since is applied in both sides, we can cancel it leading precisely to the initial equality that we wanted to demonstrate (Eq. (3.54)).

Next, substituting this relation in the volume averaged gradient given in Eq. (3.53) leads to

$$[\nabla f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} \nabla f(\boldsymbol{r}^*)\,\mathrm{d}\mathcal{V}^* = \frac{1}{\mathcal{V}_a} \int_{\mathcal{S}_a} \boldsymbol{r}^*\, \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* - \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} \boldsymbol{r}^*\, \nabla^2 f(\boldsymbol{r}^*)\mathrm{d}\mathcal{V}^* \,. \tag{B.7}$$

Let's focus on the surface integral in the right hand side. We can subdivide it as a summation of integrals, one for each face, where the normal vector is constant, $\boldsymbol{e}_{ba}$, as discussed in Sec. 3.2.1.

$$\int_{\mathcal{S}_a} \boldsymbol{r}^* \, \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* = \sum_{b \neq a} \int_{\mathcal{S}_{ab}} \boldsymbol{r}^* \, \nabla f(\boldsymbol{r}^*) \cdot \boldsymbol{e}_{ba} \mathrm{d}\mathcal{S}^* \,. \tag{B.8}$$

Furthermore, since the Voronoi face is placed at half way between the particles, we can express any point on it as

$$\boldsymbol{r}^* = \boldsymbol{r}_a + \frac{1}{2} \boldsymbol{r}_{ba} + \boldsymbol{r}_\perp^* \tag{B.9}$$

where $\boldsymbol{r}_\perp^*$ is a new integration variable that takes the midpoint between a and b as the origin, as shown in Fig. B.1.

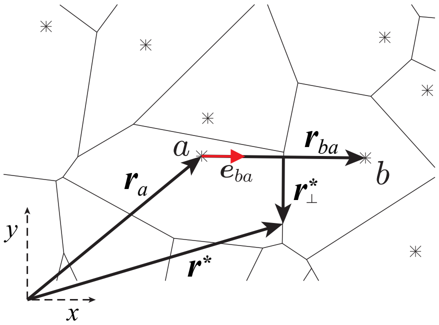


Figure B.1: Voronoi vector nomenclature

Substituting this expansion of $\boldsymbol{r}^*$ into the surface integral leads to

$$\int_{\mathcal{S}_a} \boldsymbol{r}^* \, \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* = \boldsymbol{r}_a \sum_{b \neq a} \int_{\mathcal{S}_{ab}} \nabla f(\boldsymbol{r}^*) \cdot \boldsymbol{e}_{ba} \mathrm{d}\mathcal{S}^* + \sum_{b \neq a} \int_{\mathcal{S}_{ab}} \left(\frac{\boldsymbol{r}_{ba}}{2} + \boldsymbol{r}_\perp^* \right) \nabla f(\boldsymbol{r}^*) \cdot \boldsymbol{e}_{ba} \mathrm{d}\mathcal{S}^* \,. \tag{B.10}$$

The first term of the RHS in this expression can be traced back to a single integral through all the faces, and using the divergence theorem directly converted into a volume integral:

$$\boldsymbol{r}_a \sum_{b \neq a} \int_{\mathcal{S}_{ab}} \nabla f(\boldsymbol{r}^*) \cdot \boldsymbol{e}_{ba} \mathrm{d}\mathcal{S}^* = \boldsymbol{r}_a \int_{\mathcal{S}_a} \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* = \int_{\mathcal{V}_a} \boldsymbol{r}_a \nabla^2 f(\boldsymbol{r}^*) \mathrm{d}\mathcal{V}^* \,. \tag{B.11}$$

Regarding the second term, we first approximate the gradient as constant over the face and determine it assuming a linear variation of the field values between the particles:

$$\nabla f(\boldsymbol{r}^*) \approx \frac{f(\boldsymbol{r}_b) - f(\boldsymbol{r}_a)}{||\boldsymbol{r}_b - \boldsymbol{r}_a||} \boldsymbol{e}_{ba} = \frac{f_{ba}}{||\boldsymbol{r}_{ba}||} \boldsymbol{e}_{ba} \,, \tag{B.12}$$

acceptable if the fields are sufficiently smooth. Using this approximation in the second integral of Eq. (B.10), and since $\boldsymbol{e}_{ba} = \boldsymbol{r}_{ba} / ||\boldsymbol{r}_{ba}||$ and $\boldsymbol{e}_{ba} \cdot \boldsymbol{e}_{ba} = 1$, it transforms into

$$\int_{\mathcal{S}_{ab}} \left(\frac{\boldsymbol{r}_{ba}}{2} + \boldsymbol{r}_\perp^* \right) \frac{f_{ba}}{||\boldsymbol{r}_{ba}||} \boldsymbol{e}_{ba} \cdot \boldsymbol{e}_{ba} \mathrm{d}\mathcal{S}^* = f_{ba} \int_{\mathcal{S}_{ab}} \left(\frac{\boldsymbol{e}_{ba}}{2} + \frac{\boldsymbol{r}_\perp^*}{||\boldsymbol{r}_{ba}||} \right) \mathrm{d}\mathcal{S}^* \,. \tag{B.13}$$

Furthermore,

$$\int_{\mathcal{S}_{ab}} \frac{\boldsymbol{e}_{ba}}{2}\, \mathrm{d}\mathcal{S}^* = \frac{\boldsymbol{e}_{ba}}{2} \int_{\mathcal{S}_{ab}} \mathrm{d}\mathcal{S}^* = \frac{\boldsymbol{e}_{ba}}{2} A_{ab}\,, \tag{B.14}$$

$$\int_{\mathcal{S}_{ab}} \frac{\boldsymbol{r}_\perp^*}{\|\boldsymbol{r}_{ba}\|}\, \mathrm{d}\mathcal{S}^* = \frac{1}{\|\boldsymbol{r}_{ba}\|} \int_{\mathcal{S}_{ab}} \boldsymbol{r}_\perp^*\, \mathrm{d}\mathcal{S}^* = \frac{\boldsymbol{c}_{ba}}{\|\boldsymbol{r}_{ba}\|} A_{ab}\,, \tag{B.15}$$

where $\boldsymbol{c}_{ba}$ is the position vector of the centroid of the face area relative to the middle point between a and b,[1] since

$$\boldsymbol{c}_{ba} = \boldsymbol{c}_{ab} = \boldsymbol{r}^*_{\text{centroid},\, ab} - \frac{\boldsymbol{r}_a + \boldsymbol{r}_b}{2} = \frac{\int_{\mathcal{S}_{ab}} \boldsymbol{r}^* - \left(\frac{\boldsymbol{r}_a+\boldsymbol{r}_b}{2}\right) \mathrm{d}\mathcal{S}^*}{\int_{\mathcal{S}_{ab}} \mathrm{d}\mathcal{S}^*} = \frac{\int_{\mathcal{S}_{ab}} \boldsymbol{r}_\perp^*\, \mathrm{d}\mathcal{S}^*}{A_{ab}}\,. \tag{B.16}$$

Using these expressions in the RHS of Eq. (B.10), we can rewrite it as

$$\int_{\mathcal{S}_a} \boldsymbol{r}^*\, \nabla f(\boldsymbol{r}^*) \cdot \mathbf{d}\boldsymbol{\mathcal{S}}^* = \int_{\mathcal{V}_a} \boldsymbol{r}_a\, \nabla^2 f(\boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* + \sum_{b\neq a} A_{ab}\, f_{ba} \left(\frac{\boldsymbol{e}_{ba}}{2} + \frac{\boldsymbol{c}_{ba}}{\|\boldsymbol{r}_{ba}\|} \right). \tag{B.17}$$

Finally, substituting it in the original expansion for the volume-averaged gradient in Eq. (B.7), and merging the two volume integrals, we arrive at

$$[\nabla f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \sum_{b\neq a} A_{ab}\, f_{ba} \left(\frac{\boldsymbol{e}_{ba}}{2} + \frac{\boldsymbol{c}_{ba}}{\|\boldsymbol{r}_{ba}\|} \right) + \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} (\boldsymbol{r}_a - \boldsymbol{r}^*)\, \nabla^2 f(\boldsymbol{r}^*) \mathrm{d}\mathcal{V}^*\,. \tag{B.18}$$

The following observations can be made:

- For linear fields, $\nabla^2 f(\boldsymbol{r}^*) = 0$ and the second integral vanishes. Furthermore, the approximation of the gradient in Eq. (B.12) is exact. Therefore, this volume integral is a second order correction, and the derived expression for ∇f is **exact for uniform and linear fields, independently of the particle arrangement.**

- If we still consider second order values, we can approximate $\nabla^2 f(\boldsymbol{r}^*)$ as a constant value, $\nabla^2 f_a$, so the volume integral can be approximated to a third order as

  $$\frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} (\boldsymbol{r}_a - \boldsymbol{r}^*)\, \nabla^2 f(\boldsymbol{r}^*) \mathrm{d}\mathcal{V}^* \approx \nabla^2 f_a \frac{\int_{\mathcal{V}_a} (\boldsymbol{r}_a - \boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^*}{\mathcal{V}_a} = \nabla^2 f_a\, (\boldsymbol{r}_a - \boldsymbol{r}_{\text{centroid},\, a}) \tag{B.19}$$

  We conclude that as long as the particles are close to the centroid of the Voronoi cells and the fields are sufficiently smooth, this second order correction is negligible, arriving at the final expression for $[\nabla f(\boldsymbol{r})]_a = \nabla f_a$ used in VPH, Eq. (3.58):

  $$\nabla f_a = \frac{1}{\mathcal{V}_a} \sum_{b\neq a} A_{ab}\, f_{ab} \left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right]$$

  where the relations $f_{ba} = -f_{ab}$, $\boldsymbol{e}_{ba} = -\boldsymbol{e}_{ab}$, and $\boldsymbol{c}_{ba} = \boldsymbol{c}_{ab}$ have been used.

---

[1] Remember that the origin of the relative position vector $\boldsymbol{r}_\perp^*$ is the middle point between a and b

- Note on accuracy: If the particles drift away from the centroid, the error due to neglecting these second order term may become significant. This fact explains the advantage of introducing a shifting algorithm to ensure that the particles do not deviate from the centroids more than a selected threshold. Despite introducing non-physical particle movements, *i.e.* not due to any force, the avoidance of the loss of accuracy of the volume averaged approximations is more beneficial overall.

APPENDIX C

NUMERICAL DAMPING IN VPH

When no viscous dissipation forces are included in the model, the explicit VPH formulation leads to an unstable scheme. Thus, we need to introduce an artificial viscous component to ensure that we dissipate the fictitious energy introduced numerically in the system. We can visualize this by analyzing the simple uni-dimensional[1] configuration shown in Fig. C.1, with only two particles, bounded at the bottom, and free at the top.

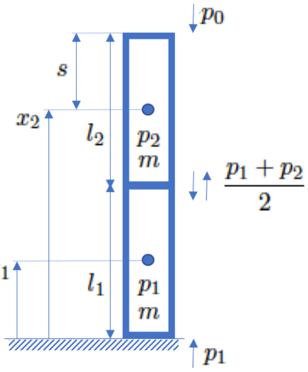


Figure C.1: Schematic of a simple two-particle configuration

We can establish the following relations for a weakly compressible fluid where we determine the density from the particle cell size:

$$\rho_1 = \frac{m}{l_1}\,, \quad \rho_2 = \frac{m}{l_2}\,, \quad s = \frac{l_0}{2}\,, \tag{C.1}$$

$$x_1 = l_1 - l_2 + s\,, \tag{C.2}$$

$$x_2 = l_1 + l_2 - s\,, \tag{C.3}$$

$$\frac{p_1 + p_2}{2} = \frac{c^2\rho_0}{2\gamma}\left[\left(\frac{m}{\rho_0 l_1}\right)^{\gamma} + \left(\frac{m}{\rho_0 l_2}\right)^{\gamma} - 2\right] + p_0\,, \tag{C.4}$$

$$\frac{p_1 - p_2}{2} = \frac{c^2\rho_0}{2\gamma}\left[\left(\frac{m}{\rho_0 l_1}\right)^{\gamma} - \left(\frac{m}{\rho_0 l_2}\right)^{\gamma}\right]. \tag{C.5}$$

C.1 NO DAMPING CASE

Considering only the pressure component, the equilibrium of forces on the system leads to

$$\sum F_1 = m\ddot{x}_1 = p_1 - \frac{p_1 + p_2}{2} = \frac{p_1 - p_2}{2}\,, \tag{C.6}$$

$$\sum F_2 = m\ddot{x}_2 = -p_0 + \frac{p_1 + p_2}{2}\,. \tag{C.7}$$

---

[1] In 1D, Voronoi cells are segments, with the limit between cells at half way between the particles

Stating p and x as functions of l, we arrive at the differential system of equations in l_1 and l_2 that define the movement of the system, which can be simplified by adding and subtracting the initial set of equations.

$$\begin{cases} m\ddot{l}_1 - m\ddot{l}_2 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_1}\right)^\gamma - \left(\dfrac{m}{\rho_0 l_2}\right)^\gamma\right] \\ m\ddot{l}_1 + m\ddot{l}_2 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_1}\right)^\gamma + \left(\dfrac{m}{\rho_0 l_2}\right)^\gamma - 2\right] - p_0 \end{cases} \rightarrow \begin{cases} m\ddot{l}_1 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_1}\right)^\gamma - 1\right] - \dfrac{p_0}{2} \\ m\ddot{l}_2 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_2}\right)^\gamma - 1\right] + \dfrac{p_0}{2}\,. \end{cases} \tag{C.8}$$

The result is a non-coupled system of nonlinear ODEs. Looking at l_1, we can rewrite it as

$$\ddot{l}_1 - \frac{c^2 m^{\gamma-1}}{2\gamma\rho_0^{\gamma-1}} l_1^{-\gamma} + \frac{c^2\rho_0}{2\gamma m} + \frac{p_0}{2m} = 0\,. \tag{C.9}$$

No large variations of l are expected since we are dealing with a weakly compressible fluid. As a first approximation, we can Taylor expand the second term around the equilibrium length, l_0, leading to

$$l_1^{-\gamma} \approx -\gamma l_0^{-\gamma-1}(l_1 - l_0)\,, \tag{C.10}$$

which substituted in the ODE, and noticing that $\frac{m}{\rho_0} = l_0$, leads to its linearized form:

$$\ddot{l}_1 + \left(\frac{c^2}{l_0^2}\right) l_1 = \left(\frac{c^2}{2l_0} - \frac{c^2}{2\gamma l_0} - \frac{p_0}{2m}\right)\,, \tag{C.11}$$

The analytic solution of this ODE has complex eigenvalues with zero real parts, *i.e.* consists in pure oscillations around an equilibrium length. Its solution using an explicit numerical scheme is unstable, confirming our initial statement.

C.2 Damping included

The first way to obtain a stable system is include damping as was initially devised. Inspired by the stress tensor for Newtonian fluids,

$$\underline{\underline{\sigma}} = -p\mathbb{I} + \lambda\,(\nabla \cdot \boldsymbol{u})\,\mathbb{I} + \mu\, 2\,\mathbb{D}\,,$$

where $\mathbb{I}$ is the identity and $\mathbb{D}$ the rate of deformation tensors, we can simply include a new damping term proportional to the divergence of the velocity,

$$f_\beta = \beta\,\nabla \cdot \boldsymbol{v}\,, \tag{C.12}$$

since in 1D

$$\nabla \cdot \boldsymbol{u} = 2\,\mathbb{D} = \frac{\mathrm{d}\,\boldsymbol{u}}{\mathrm{d}\,x} \approx \frac{\Delta\dot{x}}{\Delta x}\,. \tag{C.13}$$

Evaluated on each cell limit yields:

$$\nabla \cdot \boldsymbol{u}\,|_{x=0} = \frac{\dot{x}_1}{x_1}\,, \tag{C.14}$$

$$\nabla \cdot \boldsymbol{u}\,|_{x=l_1} = \frac{\dot{x}_2 - \dot{x}_1}{x_2 - x_1}\,, \tag{C.15}$$

$$\nabla \cdot \boldsymbol{u}\,|_{x=l_1+l_2} = 0\,. \tag{C.16}$$

Substituting them on the original equilibrium of forces, and taking into account that the force acts along the normal to each face pointing outwards, we obtain:

$$\sum F_1 = m\ddot{x}_1 = \frac{p_1 - p_2}{2} - \beta\frac{\dot{x}_1}{x_1} + \beta\frac{\dot{x}_2 - \dot{x}_1}{x_2 - x_1}\,, \tag{C.17}$$

$$\sum F_2 = m\ddot{x}_2 = \frac{p_1 + p_2}{2} - p_0 - \beta\frac{\dot{x}_2 - \dot{x}_1}{x_2 - x_1}\,. \tag{C.18}$$

Following the same steps than before, we arrive at the final system of ODEs for this case

$$\begin{cases} m\ddot{l}_1 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_1}\right)^{\gamma} - 1\right] - \dfrac{p_0}{2} - \dfrac{\beta}{2}\dfrac{\dot{l}_1 - \dot{l}_2}{l_1 - l_2 + s}\,, \\ m\ddot{l}_2 = \dfrac{c^2\rho_0}{2\gamma}\left[\left(\dfrac{m}{\rho_0 l_2}\right)^{\gamma} - 1\right] + \dfrac{p_0}{2} + \dfrac{\beta}{2}\dfrac{\dot{l}_1 - \dot{l}_2}{l_1 - l_2 + s} - \beta\dfrac{\dot{l}_2}{l_2 - s}\,. \end{cases} \tag{C.19}$$

Based on the expected small variations of l, we again use the Taylor expansion $l_1^{-\gamma} \approx -\gamma l_0^{-\gamma-1}(l_1 - l_0)$. Furthermore, we can assume that

$$l_1 - l_2 + s \approx s = \frac{l_0}{2}\,, \tag{C.20}$$

since both l_1 and l_2 are both on the order of l_0, so its difference is negligible compared to s. We can also use this fact to simplify the last coefficient in the second ODE, by assuming

$$l_2 - s \approx l_0 - \frac{l_0}{2} = \frac{l_0}{2}\,. \tag{C.21}$$

Under all these assumptions, the linearized form of our system of ODEs are

$$\begin{cases} \ddot{l}_1 = -\dfrac{\beta}{ml_0}\dot{l}_1 + \dfrac{\beta}{ml_0}\dot{l}_2 - \left(\dfrac{c^2}{l_0^2}\right) l_1 + \left(\dfrac{c^2}{2l_0} - \dfrac{c^2}{2\gamma l_0} - \dfrac{p_0}{2m}\right) \\ \ddot{l}_2 = +\dfrac{\beta}{ml_0}\dot{l}_1 - \dfrac{3\beta}{ml_0}\dot{l}_2 - \left(\dfrac{c^2}{l_0^2}\right) l_2 + \left(\dfrac{c^2}{2l_0} - \dfrac{c^2}{2\gamma l_0} + \dfrac{p_0}{2m}\right) \end{cases} \tag{C.22}$$

The stability of the system depends on the coefficients multiplying the particle lengths and its derivatives in the LHS. Therefore, without any loss of generality we will assume $\gamma = 1$ and $p_0 = 0$, so all constant values on the r values become zero. The system of ODE remain coupled, but if we define an auxiliary variable $\varsigma = \dot{l}$, we can rewrite it as

$$\begin{bmatrix} \dot{\varsigma}_1 \\ \dot{\varsigma}_2 \\ \dot{l}_1 \\ \dot{l}_2 \end{bmatrix} = \begin{bmatrix} -\frac{\beta}{ml_0} & +\frac{\beta}{ml_0} & -\frac{c^2}{l_0^2} & 0 \\ +\frac{\beta}{ml_0} & -\frac{3\beta}{ml_0} & 0 & -\frac{c^2}{l_0^2} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varsigma_1 \\ \varsigma_2 \\ l_1 \\ l_2 \end{bmatrix} \quad \rightarrow \quad \dot{\boldsymbol{\ell}} = \mathbb{J}\,\boldsymbol{\ell} \tag{C.23}$$

The stability of the system depends on the eigenvalues λ of the Jacobian matrix $\mathbb{J}$.

$$|\mathbb{J} - \lambda\mathbb{I}| = \lambda^4 + \lambda^3\left(\frac{4\beta}{ml_0}\right) + \lambda^2\left(\frac{2\beta^2}{m^2 l_0^2} + \frac{2c^2}{l_0^2}\right) + \lambda\left(\frac{4\beta c^2}{m l_0^3}\right) + \frac{c^4}{l_0^4} = 0\,. \tag{C.24}$$

Defining

$$A = \frac{c^2}{l_0^2}\,, \quad B = \frac{\beta}{ml_0}\,, \tag{C.25}$$

we can determine the eigenvalues as

$$\lambda_{1,2} = \pm\sqrt{\frac{-2A + (3 - 2\sqrt{2})B^2}{2}} + \frac{1 - \sqrt{2}}{\sqrt{2}}B\,, \tag{C.26}$$

$$\lambda_{3,4} = \pm\sqrt{\frac{-2A + (3 + 2\sqrt{2})B^2}{2}} + \frac{-1 - \sqrt{2}}{\sqrt{2}}B\,. \tag{C.27}$$

Noticing that $(3 - 2\sqrt{2}) = (1 - \sqrt{2})^2$ and $(3 + 2\sqrt{2}) = (-1 - \sqrt{2})^2$, we can further simplify these expressions by defining

$$B_1 = (1 - \sqrt{2})B\,, \quad B_2 = (-1 - \sqrt{2})B\,, \tag{C.28}$$

leading to

$$\sqrt{2}\,\lambda_{1,2} = \pm\sqrt{-2A + B_1^2} + B_1\,, \tag{C.29}$$

$$\sqrt{2}\,\lambda_{3,4} = \pm\sqrt{-2A + B_2^2} + B_2\,. \tag{C.30}$$

For $\beta > 0 \implies B > 0 \implies B_1 < 0$ and $B_2 < 0$, the following conditions are extracted to ensure that the real parts of the eigenvalues are negative, condition for the system to be stable:

$$\begin{cases} |B_1| > +\sqrt{-2A + B_1^2} \\ |B_2| > +\sqrt{-2A + B_2^2} \end{cases} \tag{C.31}$$

Since $A > 0$

$$\begin{cases} B_1^2 - 2A < B_1^2 \implies +\sqrt{B_1^2 - 2A} < |B_1| \quad \forall B_1 \\ B_2^2 - 2A < B_2^2 \implies +\sqrt{B_2^2 - 2A} < |B_2| \quad \forall B_2\,. \end{cases} \tag{C.32}$$

We conclude that as long as we introduce some damping, the VPH explicit scheme is stable if an appropriate time step is chosen (Sec. 4.5).

C.3 ORDER OF MAGNITUDE OF NUMERICAL DAMPING

We have seen how any $\beta > 0$ is valid. However, we should keep in mind that the artificial viscosity is a fictitious component. Therefore, we should keep it as low as possible so its impact on the flow fields is negligible, *i.e.* the deviations for the physical response are not noticeable.

We can achieve this by determining the order of magnitude of the pressure forces, F_p and making the numerical damping one, F_β, proportional to it:

$$F_\beta = \alpha F_p \,, \tag{C.33}$$

where we choose α to ensure that F_β is at least one order of magnitude smaller than F_p to have a minimum impact in the flow, independently of the absolute value of F_p. From Eq. (2.26) we determined

$$\Delta p \approx c^2 \rho_0 \eta \,. \tag{C.34}$$

where $\eta = \frac{\Delta\rho}{\rho}$. On the other hand, using the continuity equation we can approximate the order of magnitude of the numerical damping as

$$F_\beta = \beta \nabla \cdot \boldsymbol{u} = -\beta \frac{1}{\rho} \frac{\mathrm{d}\rho}{\mathrm{d}t} \approx -\beta \frac{1}{\rho} \frac{\Delta\rho}{\Delta t} = -\beta \frac{\eta}{\Delta t} \,. \tag{C.35}$$

Based on Sec. 4.5, we can approximate the time step as $\Delta t \approx \frac{l_0}{c}$, which leads to

$$F_\beta \approx -\beta \frac{c\,\eta}{l_0} \,. \tag{C.36}$$

Substituting these expressions in Eq. (C.33), we can finally determine an expression for β

$$\beta = \alpha\, l_0\, c\, \rho_0 \,. \tag{C.37}$$

Noticing that $l_0 \approx \Delta x \approx h$, we can directly identify this value as the one used in the artificial viscosity formulation used in SPH (Eq. (3.14)) and VPH (Eq. (3.71)). Typically, $\alpha \approx 0.02$.

APPENDIX D

GEOMETRY

This appendix summarizes several geometrical properties that we can compute using the Voronoi tessellation data as described in Sec. 3.2.1 (nodes, faces, and cells). The following notation is followed unless otherwise stated:

- D indicate the number of dimensions of the geometrical space Ω
- Cartesian coordinate system assumed
- Sub-indexes used to identify vectors, and super-indexes to identify vector components. Thus, $\boldsymbol{v}_j^k$ is the k^{th} component of the j^{th} face vector.
- $\boldsymbol{x}$ and $\boldsymbol{r}$ are Voronoi nodes and fluid particle position vectors, respectively. Algebraically, vectors are treated as arrays of one column and D rows.
- $\mathcal{X}_{ab} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_D\}$ is the subset of Voronoi nodes in a cell face between particles a and b

D.1 VOLUME OF A D-DIMENSIONAL TRIANGULAR ELEMENT

A D-dimensional triangle, also named D-simplex, is defined as the domain enclosed by flat sides (polytope) with the least number of nodes in a D-dimensional space, *i.e.* $D+1$ nodes. Therefore in 1D is a segment, in 2D a triangle, or in 3D a tetrahedron for example. Its volume can be computed as a function of the nodes coordinates as

$$\begin{gathered} \boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1 \quad i = 1 \dots D\,, \\ \mathcal{V} = \frac{1}{D!}\,\mathrm{abs}\left(\begin{vmatrix} \boldsymbol{v}_1^1 & \cdots & \boldsymbol{v}_D^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_1^D & \cdots & \boldsymbol{v}_D^D \end{vmatrix} \right). \end{gathered} \tag{D.1}$$

Note that depending on the order of the nodes, the determinant can be positive or negative, hence the need to take the absolute value. Applied to a Voronoi diagram, the volume of each cell can be taken as the summation of a set of simplexes, each one delimited by the nodes of one face and the particle itself as shown in Fig. D.1(a).

$$\mathcal{V}_a = \frac{1}{D!} \sum_{b \neq a} s_{ab}\,\mathrm{abs}\left(\begin{vmatrix} (\boldsymbol{x}_1 - \boldsymbol{r}_a)^1 & \cdots & (\boldsymbol{x}_D - \boldsymbol{r}_a)^1 \\ \vdots & \ddots & \vdots \\ (\boldsymbol{x}_1 - \boldsymbol{r}_a)^D & \cdots & (\boldsymbol{x}_D - \boldsymbol{r}_a)^D \end{vmatrix} \right), \tag{D.2}$$

where s_{ab} is a coefficient introduced to account for possible concave faces resulting from the intersection with concave boundaries[1]. As shown in Fig. D.1(b), if we split the domain by the hyperplane containing the boundary face, the volume should be added to the total volume if the particle

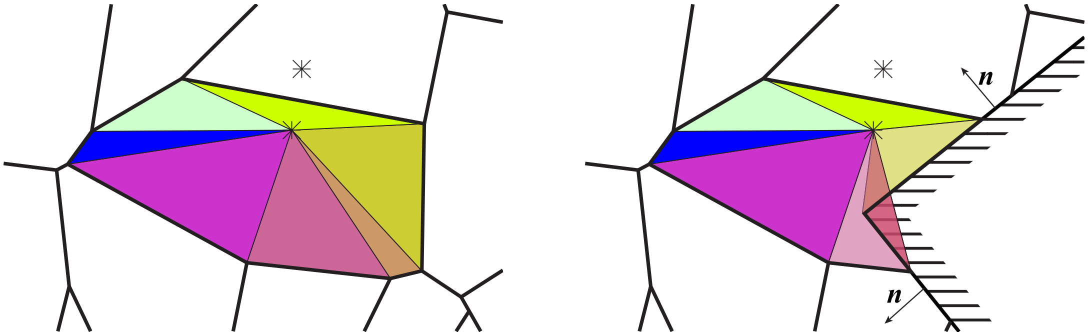
---

[1] Concave sides should be only on the boundaries, since pure Voronoi cells are always convex.

falls within the fluid side of the face ($s_{ab} = 1$), and subtracted if not ($s_{ab} = -1$). Based on the unit normal vector to the boundary, $\boldsymbol{n}$, which is always oriented towards the fluid (Appendix D.3), we can define

$$s_{ab} = \begin{cases} +1 & (\boldsymbol{r}_a - \boldsymbol{x}_1) \cdot \boldsymbol{n} \geq 0 \\ -1 & (\boldsymbol{r}_a - \boldsymbol{x}_1) \cdot \boldsymbol{n} < 0\,, \end{cases} \tag{D.3}$$

being $\boldsymbol{x}_1$ the first of node in $\mathcal{X}_{ab}$.



(a) Convex cell

(b) Concave cell

Figure D.1: Voronoi cell volume

D.2 AREA

The volume of a D-simplex can be also calculated as

$$\mathcal{V} = \frac{1}{D} A\, H\,, \tag{D.4}$$

where A is the area of the base and H its height. We can directly apply this fact to determine the area of the Voronoi cell face by using it as a base to construct a simplex with height $H = 1$. This is simply achieved if we include a vector of length 1 orthogonal to the hyper-plane containing the face, *i.e.*, its unit normal vector $\boldsymbol{n}$. In general, $\boldsymbol{n}$ can be computed based on the face nodes as shown in Appendix D.3, although we can use the fact that the Voronoi faces are orthogonal to the vector joining their related particles, so $\boldsymbol{n} = \boldsymbol{r}_{ab}/\|\boldsymbol{r}_{ab}\| = \boldsymbol{e}_{ab}$. As a result,

$$A_{ab} = = \frac{1}{(D-1)!} \operatorname{abs} \left(\begin{vmatrix} \boldsymbol{n}^1 & (\boldsymbol{x}_2 - \boldsymbol{x}_1)^1 & \cdots & (\boldsymbol{x}_D - \boldsymbol{x}_1)^1 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{n}^D & (\boldsymbol{x}_2 - \boldsymbol{x}_1)^D & \cdots & (\boldsymbol{x}_D - \boldsymbol{x}_1)^D \end{vmatrix} \right). \tag{D.5}$$

D.3 FACE UNIT NORMAL VECTOR

Both boundary faces and Voronoi cell faces are planar $D-1$ dimensional triangles. We can therefore determine the unit normal vector to them using the normalized exterior product between the

$D-1$ vectors resulting from connecting the first node of the face with the rest:

$$\boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1 \quad i = 1 \dots D-1\,,$$

$$\boldsymbol{n}^* = \boldsymbol{v}_1 \wedge \dots \wedge \boldsymbol{v}_{D-1} = \begin{vmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_n \\ \boldsymbol{v}_1^1 & \cdots & \boldsymbol{v}_1^D \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_{D-1}^1 & \cdots & \boldsymbol{v}_{D-1}^D \end{vmatrix}\,, \tag{D.6}$$

$$\boldsymbol{n} = \frac{\boldsymbol{n}^*}{\|\boldsymbol{n}^*\|}\,, \tag{D.7}$$

where $\boldsymbol{e}_i$ is the unit vector of the orthogonal basis in the i^{th} direction.

Based on this definition, a requirement is set to order each boundary face nodes in its triangulation so its normal vector points towards the fluid, *i.e.* ordered counterclockwise.

D.4 ORTHOGONAL DISTANCE FROM A POINT TO A FACE PLANE

Is obtained projecting the vector connecting the point to one of the face nodes to the face unit normal.

$$d_\perp = (\boldsymbol{r}_a - \boldsymbol{x}_1) \cdot \boldsymbol{n}\,. \tag{D.8}$$

D.5 MINIMUM DISTANCE FROM A POINT TO A FACE

The orthogonal distance to the face plane in Eq. (D.8) is not necessary the minimum distance, since the closest point of the plane may not fall on the actual face polygon. The problem of finding the minimum distance is equivalent to finding the closest point inside the face to that orthogonal projection point on the face plane.

Each face consists on a closed polygon within a hyper-plane and is defined by D points. This hyper-plane represents a subspace of the domain with dimension $D-1$. Taking the first node of the polygon as the origin, $\boldsymbol{x}_1$, we can define $D-1$ vectors $\boldsymbol{v}_i$ like before, which conform a base of this subspace (assuming that the face is not ill-defined, so there are no coincident nodes)

$$\boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1 \quad i = 1 \dots D-1\,. \tag{D.9}$$

In order to determine the minimum distance, we need to determine the closest point within the face, $\boldsymbol{x}_c$, to the analyzed point, $\boldsymbol{r}_a$. This point belongs to the hyper-plane subspace, so it can be defined as a linear combination of the base vectors $\boldsymbol{v}_i$.

$$\boldsymbol{x}_c = \boldsymbol{x}_1 + \prod_{i=1}^{D-1} \lambda_i \boldsymbol{v}_i = \boldsymbol{x}_1 + \begin{bmatrix} \boldsymbol{v}_1^1 & \cdots & \boldsymbol{v}_{D-1}^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_1^D & \cdots & \boldsymbol{v}_{D-1}^D \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^1 \\ \vdots \\ \boldsymbol{\lambda}^{D-1} \end{bmatrix} = \boldsymbol{x}_1 + V\boldsymbol{\lambda}\,. \tag{D.10}$$

The square of the distance, d, is then given by

$$\begin{aligned} d^2 &= \|\boldsymbol{x}_c - \boldsymbol{r}_a\|^2 = \|\boldsymbol{x}_1 - \boldsymbol{r}_a + V\boldsymbol{\lambda}\|^2 = \|\boldsymbol{r} + V\boldsymbol{\lambda}\|^2 = (\boldsymbol{r} + V\boldsymbol{\lambda}) \cdot (\boldsymbol{r} + V\boldsymbol{\lambda}) \\ &= \boldsymbol{r}^T\boldsymbol{r} + \boldsymbol{r}^T(V\boldsymbol{\lambda}) + (V\boldsymbol{\lambda})^T\boldsymbol{r} + (V\boldsymbol{\lambda})^T(V\boldsymbol{\lambda})\,, \end{aligned} \tag{D.11}$$

where ^{T} indicate the transpose. Using the property that $\boldsymbol{v} = (\boldsymbol{v}^T)^T$, and that each of the elements of the previous expression lead to an scalar value, so $\boldsymbol{a}\boldsymbol{b}^T = (\boldsymbol{a}\boldsymbol{b}^T)^T = \boldsymbol{b}\boldsymbol{a}^T$, we can rewrite the previous expression as

$$d^2 = \boldsymbol{r}^T\boldsymbol{r} + 2\boldsymbol{r}^T V\boldsymbol{\lambda} + \boldsymbol{\lambda}^T V^T V\boldsymbol{\lambda}\,. \tag{D.12}$$

We are looking for the minimum distance, which should satisfy the condition that the derivative is zero, or what is equivalent, that the derivative of d^2 is zero. Therefore

$$\frac{\mathrm{d}\,d^2}{\mathrm{d}\boldsymbol{\lambda}} = 2\boldsymbol{r}^T V + (V^T V\boldsymbol{\lambda})^T + \boldsymbol{\lambda}^T V^T V = 2\boldsymbol{r}^T V + 2\boldsymbol{\lambda}^T V^T V = 0\,. \tag{D.13}$$

Taking the transpose of the complete expression, we can rewrite it leading to a algebraic system of equations whose solution provides the values of $\boldsymbol{\lambda}^i$ that determine the location of the closest point in the hyper-plane.

$$V^T V\boldsymbol{\lambda} = -V^T\boldsymbol{r}\,. \tag{D.14}$$

However, the resulting point may not lay within the face limits if it doesn't satisfy the conditions stated in Appendix D.6. To locate the closest point within the face we can do it recursively as:

1. If exist values of $\boldsymbol{\lambda}^i < 0$, set them to zero and as they lead to points away from the face. If not, go to step 3.
2. Recompute with the remaining valid vectors, $\left\{\boldsymbol{v}_i \,|\, \boldsymbol{\lambda}^i \geq 0\right\}$ the new values of $\boldsymbol{\lambda}^i$. If some of the new values of $\boldsymbol{\lambda}^i$ are negative, go back to the previous state and iterate until all the values are positive.
3. Check that the condition of $\sum_{i=1}^{D-1} \boldsymbol{\lambda}^i \leq 1$ is satisfied. If so, a valid point has been found. Otherwise, we scale the values of λ_i to satisfy it

$$\boldsymbol{\lambda}^i = \frac{\boldsymbol{\lambda}^{i\,(*)}}{\sum_{i=1}^{D-1} \boldsymbol{\lambda}^{i\,(*)}}\,, \tag{D.15}$$

where $\boldsymbol{\lambda}^{i\,(*)}$ represent the original values.

D.6 Point inside a generalized triangular element

We can always express any point $\boldsymbol{x}$ inside of a D-dimensional triangle as a linear combination of the base vectors obtained by connecting the first node to all the rest,

$$\mathcal{B} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D \,|\, \boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1\}\,, \tag{D.16}$$

so

$$\boldsymbol{x} = \lambda^1 \boldsymbol{v}_1 + \ldots + \lambda^n \boldsymbol{v}_n = \begin{bmatrix} \boldsymbol{v}_1^1 & \cdots & \boldsymbol{v}_D^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}_1^D & \cdots & \boldsymbol{v}_D^D \end{bmatrix} \begin{bmatrix} \lambda^1 \\ \vdots \\ \lambda^D \end{bmatrix} . \tag{D.17}$$

Note that $\boldsymbol{\lambda} = [\lambda^1; \ldots; \lambda^D]$ are the node coordinates in the new base, although the basis itself is not orthogonal. The conditions for a point to belong to the triangular element are:

$$\lambda^i \geq 0 \quad \forall i = 1 \ldots D , \tag{D.18}$$

$$\sum_{i=1}^{D} \lambda^i \leq 1 . \tag{D.19}$$

In addition, a point is located on one of the sides of the triangle if one of the conditions below are satisfied:

$$\exists \lambda^i = 0 , \tag{D.20}$$

$$\sum_{i=1}^{D} \lambda^i = 1 . \tag{D.21}$$

Finally, the point is located along one of the edges connecting two nodes if the condition below is satisfied,

$$\begin{cases} 0 \leq \lambda_j \leq 1 \\ \lambda_i = 0 \quad \forall i \neq j . \end{cases} \tag{D.22}$$

As a final remark, all these conditions can be directly extended to a lower dimension simplex within the D-dimensional space by simply replacing D with the dimension order of the simplex. For example, in 3D we can evaluate if a point falls within a planar triangle by simply applying these expressions with $D = 2$, which basically is the dimension of the sub-space to which all the points in the triangle belong.

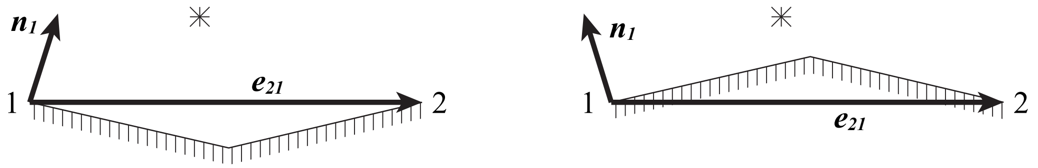
D.7 CONCAVE/CONVEX NEIGHBOR BOUNDARY FACES

Two adjacent faces are defined as those sharing $D - 1$ nodes. When seen from the fluid particles, they can be convex or concave as shown in Fig. D.2. Generally, Voronoi cells are always convex when looked from the particle inside the cell. However, when trimmed with concave boundaries may include concave adjacent faces that are treated differently, as discussed in Sec. 3.4.3.

The condition used to determine whether adjacent faces are convex or concave is based on the dot product of the vector connecting the free nodes from the first and second face[2], $\boldsymbol{x}_{\mathrm{f}1}$ and $\boldsymbol{x}_{\mathrm{f}2}$,

---

[2] Each face have D nodes. Since $D - 1$ are shared, there is only one unshared node in each face independently of the dimension of the problem



(a) Convex

(b) Concave

Figure D.2: Convex vs. concave adjacent faces in 2D

respectively, and the normal to the first face, $\boldsymbol{n}_1$, such that

$$\boldsymbol{e}_{21} = \boldsymbol{x}_{\mathrm{f}2} - \boldsymbol{x}_{\mathrm{f}1}\,, \tag{D.23}$$

$$\begin{cases} \boldsymbol{e}_{21}\cdot\boldsymbol{n}_1 \geq 0 & \rightarrow \quad \text{convex} \\ \boldsymbol{e}_{21}\cdot\boldsymbol{n}_1 < 0 & \rightarrow \quad \text{concave}\,. \end{cases} \tag{D.24}$$

D.8 SEARCH THRESHOLD TO DETECT CONVEX AND CONCAVE BOUNDARIES

We have determined when estimating the grid size in Sec. 4.3.1 that the maximum expected variation of the inter-particle distance can be upper bounded based on the stability limitations for explicit scheme (Sec. 4.5) as

$$d^{t}_{\max a} \leq d^{t-\Delta t}_{\max a} + \frac{\max(\|\Delta\boldsymbol{r}_{ab}\|)^{t-\Delta t}}{2} \leq d^{t-\Delta t}_{\max a} + \max(d_{\max a},\, d_{\max b})^{t-\Delta t}\,.$$

When applied to convex boundaries, since their virtual points are reflections of the fluid particle,

$$d_{\max b} = d_{a\perp B} \leq d_{\max a}\,. \tag{D.25}$$

As a result, the following distance threshold ensures that we don't miss any convex boundary previously linked:

$$0 < d^{t}_{a\perp B} \leq 2\, d^{t-\Delta t}_{\max a}\,, \tag{D.26}$$

using $d^{t-\Delta t}_{\max a} \approx \frac{\Delta x_a}{2}$ if no previous value is available. However, even if this condition is satisfied, we showed in Sec. 3.4.3 that no boundary virtual points should be placed when concave faces are located in the nearby. We define “nearby” as the range when a possible virtual point can lay closer to another face, BB, than the one used to create it B:

$$|\, d_{a^B\perp BB}\,| > |\, d_{a^B\perp B}\,| = |\, d_{a\perp B}\,|\,. \tag{D.27}$$

Graphically, Fig. D.3 show how we can establish

$$|\, d_{a\perp BB}\,| = |\, d_{a^B\perp BB}\,| + |\, \boldsymbol{r}_{ab}\cdot\boldsymbol{n}_{BB}\,| \leq |\, d_{a^B\perp BB}\,| + \|\boldsymbol{r}_{ab}\| = |\, d_{a^B\perp BB}\,| + 2\,|\, d_{a\perp B}\,|\,. \tag{D.28}$$

Reordering the terms, we conclude that

$$|\, d_{a^B\perp BB}\,| \geq |\, d_{a\perp BB}\,| - 2\,|\, d_{a\perp B}\,|\,. \tag{D.29}$$

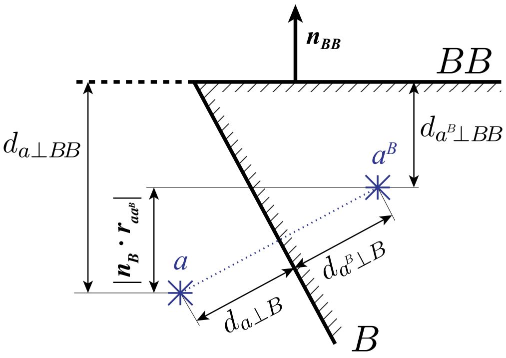


Figure D.3: Boundaries search threshold to detect concave faces

Based on Eqs. (D.26), (D.27), and (D.29), we arrive at the following threshold to link a boundary to a particle:

$$\left| d^{t}_{a\perp B} \right| \leq \begin{cases} 2\, d^{t-\Delta t}_{\max a} & \text{only convex boundaries,} \\ 6\, d^{t-\Delta t}_{\max a} & \text{otherwise.} \end{cases} \tag{D.30}$$

This definition guarantees that no boundary beyond this search scope can actually be closer to any virtual point created from a linked convex boundary, since

$$\left. \begin{array}{l} |\, d_{a\perp BB} \,| > 6\, d^{t-\Delta t}_{\max a} \\ |\, d_{a\perp B} \,| \leq 2\, d^{t-\Delta t}_{\max a} \end{array} \right\} \Longrightarrow \; |\, d_{a^B\perp BB} \,| \geq |\, d_{a\perp BB} \,| - 2\; |\, d_{a\perp B} \,| > 2\, d^{t-\Delta t}_{\max a} \geq \; |\, d_{a\perp B} \,| \tag{D.31}$$

D.9 INTERSECTION LINE AND FACE

A line is defined by a base point, $\boldsymbol{x}_l$ and a direction vector $\boldsymbol{v}$, while a face plane is defined by a face point $\boldsymbol{x}_1$ and a normal vector $\boldsymbol{n}$. The intersection point, $\boldsymbol{x}_i$ should belong to the line and the face simultaneously, so

$$\begin{cases} \boldsymbol{x}_i = \boldsymbol{x}_l + \lambda \boldsymbol{v} \\ (\boldsymbol{x}_i - \boldsymbol{x}_1) \cdot \boldsymbol{n} = 0 \end{cases} . \tag{D.32}$$

Combining both, we determine λ that gives the intersection point coordinates

$$\lambda = \frac{(\boldsymbol{x}_1 - \boldsymbol{x}_l) \cdot \boldsymbol{n}}{\boldsymbol{v} \cdot \boldsymbol{n}} \tag{D.33}$$

D.10 POINT WITHIN THE ORTHOGONAL PROJECTION OF BOUNDARY FACE

The orthogonal distance from a point to a face is given by Eq. (D.8). If we want to determine if the orthogonal projection of the point into the face plane lays within the face, we can do so in a simpler way than the procedure to determine the minimum distance described in Appendix D.5.

We have seen how each face consists of a closed polytope defined by D points, such that we can define a set of $D-1$ vectors $\boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1$ with $i = 1 \dots D-1$. We need one more vector outside of the face hyper-plane to define a valid basis for the domain space, which can be simply the normal vector to the face $\boldsymbol{n}$. Noting that we used $\boldsymbol{x}_1$ as the reference origin for the in-plane base vectors, we can express the relative position of any given point $\boldsymbol{x}_p$ to the face as $\boldsymbol{r} = \boldsymbol{x}_p - \boldsymbol{x}_1$, which also can be expressed as a combination of the base vectors. Therefore,

$$\boldsymbol{r} = \lambda_0\, \boldsymbol{n} + \prod_{i=1}^{D-1} \lambda_i \boldsymbol{v}^i = \begin{bmatrix} \boldsymbol{n}^1 & \boldsymbol{v}_1^1 & \cdots & \boldsymbol{v}_{D-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{n}^D & \boldsymbol{v}_1^D & \cdots & \boldsymbol{v}_{D-1}^D \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \vdots \\ \lambda_{D-1} \end{bmatrix} = V\boldsymbol{\lambda} \tag{D.34}$$

This consist on an algebraic system of equations that can be solved to determine the values of λ_i, which allow us to extract the following information:

- λ_0 is the orthogonal distance to the face.
- In order for $\boldsymbol{x}_p$ to be located within the orthogonal projection of the face, it needs to be satisfied that:

$$\lambda_i \geq 0 \quad \forall i = 1 \dots D-1 \tag{D.35}$$

$$\sum_{i=1}^{D-1} \lambda_i \leq 1 \tag{D.36}$$