Universidade da Coruña

Doctoral Thesis

From mesh to meshless: A generalized meshless formulation based on Riemann solvers for Computational Fluid Dynamics

Antonio Eirís Barca

Supervisors:

Dr. Xesús Antón Nogueira Garea
Dr. Luis Ramírez Palacios

Programa de doctorado en Ingeniería Civil
2022

Acknowledgments

I would like to express my deepest gratitude to my advisors Xesús Nogueira and Luis Ramírez from the first time I knock on their office door asking for a PhD proposal in Computational Fluid Dynamics. I am fully indebted with all their patience during my transition from an user of commercial CFD software to a researcher that runs its own code. Thanks for your understanding, support and guidance throughout all stages of this research. I extend the acknowledge to the rest of professors of the GMNI group for their kind words of encouragement.

A big thank-you to all the doctoral and post-doctoral students that I was honored to share the lab: Javi, Manu, Iván, Diego, Santi, Laura, Andrés y Ahmad. Thanks for creating a good working environment and also for some hilarious recess moments. In particular, I would like to acknowledge Javi for giving me support in programming and postprocessing tasks and Iván for providing the LaTeX template for manuscripts and presentations.

And finally I dedicate this thesis to my parents.

This work has been partially supported by the *Ministerio de Ciencia, Innovación y Universidades* (RTI2018-093366-B-100) of the Spanish Government and by the *Consellería de Educación e Ordenación Universitaria* of the *Xunta de Galicia*, cofinanced with FEDER funds and the *Universidade da Coruña*.

Abstract

From mesh to meshless: A generalized meshless formulation based on Riemann solvers for Computational Fluid Dynamics

This thesis deals with the development of high accuracy meshless methods for the simulation of compressible and incompressible flows. Meshless methods were conceived to overcome the constraints that mesh topology impose on traditional mesh-based numerical methods. Despite the fact that meshless methods have achieved a relative success in some particular applications, the truth is that mesh-based methods are still the preferred choice to compute flows that demand high-accuracy. Instead of assuming that meshless and mesh-based methods are groups of methods that follow independent development paths, in this thesis it is proposed to increase the accuracy of meshless methods by taking guidance of some successful techniques adopted in the mesh-based community.

The starting point for the development is inspired by the SPH-ALE scheme proposed by Vila. Especially, the flexibility of the ALE framework and the introduction of Riemann solvers are essential elements adopted. High accuracy is obtained by using the Moving Least Squares (MLS) technique. MLS serves multiple tasks in the implemented scheme: high order reconstruction of Riemann states, more accurate viscous flux evaluation and the replacement of the limited kernel approximation by MLS approximation with polynomial degree consistency by design. The stabilization of the scheme for compressible flows with discontinuities is based on *a posteriori* stabilization technique (MOOD) that introduces a great improvement compared with the traditional *a priori* flux limiters.

The MLSPH-ALE scheme is the first proposed meshless formulation that uses high order consistent MLS approximation in a versatile ALE framework. In addition, the procedure to obtain the semi-discrete formulation keeps track of a boundary term, which eases the implementation of the boundary conditions.

Another important contribution is related with the general concept of the MLSPH-ALE formulation. The MLSPH-ALE scheme is proved to be a global meshless formulation that under some particular settings provides the same semi-discrete equations that other meshless formulations published.

The MLSPH-ALE scheme has been tested for the computation of turbulent flows. The low dissipation inherent to the Riemann solver is compatible with the implicit LES

turbulent model. The proposed formulation is able to capture the energy cascade in the subsonic regime where traditional SPH formulations are reported to fail.

Resumen

Desde métodos con malla a métodos sin malla: Una formulación sin malla generalizada basada en solvers de Riemann para Dinámica de Fluidos Computacional

Esta tesis aborda el desarrollo de métodos sin malla de alta precisión para la simulación de flujos compresibles e incompresibles. Los métodos sin malla fueron creados para superar las restricciones que la conectividad de la malla impone a los métodos tradicionales. A pesar de haber alcanzado un éxito relativo en algunas aplicaciones, la realidad es que los métodos con malla siguen siendo la opción preferida para el cálculo de flujos que demandan alta precisión. En vez de asumir que métodos sin malla y con malla son grupos de métodos que siguen caminos de desarrollo independientes, en esta tesis se propone incrementar la precisión de los métodos sin malla tomando como guía algunas de las técnicas más exitosas empleadas en la comunidad de los métodos con malla.

El punto de partida para el desarrollo se inspira en el esquema SPH-ALE propuesto por Vila. De manera especial, la flexibilidad del marco de referencia ALE y la introducción de los solvers de Riemann son elementos esenciales adoptados. La alta precisión se obtiene con la técnica de Mínimos Cuadrados Móviles (MLS). MLS sirve múltiples funciones en la implementación del esquema: alto orden de reconstrucción de los estados de Riemann, evaluaciones más precisas de los flujos viscosos y reemplazo de la aproximación limitada tipo kernel por una aproximación MLS con un grado de consistencia polinómica arbitraria. La estabilización del esquema para flujos compresibles con discontinuidades se basa en una técnica de estabilización *a posteriori* (MOOD) que introduce una importante mejora con respecto a los tradicionales limitadores de flujo *a priori*.

El esquema MLSPH-ALE es la primera formulación sin malla propuesta que utiliza la aproximación MLS de alto orden en un marco de referencia ALE. Además, el procedimiento dado para obtener la forma semi-discreta realiza el seguimiento de un término en la frontera del dominio que facilita la implementación discreta de las condiciones de contorno.

Otra importante contribución está relacionada con el concepto general de la formulación MLSPH-ALE. Se ha demostrado que el esquema MLSPH-ALE es una formulación

sin malla global que con ciertas configuraciones particulares es capaz de proporcionar las mismas formas semi-discretas que otras formulaciones publicadas.

El método MLSPH-ALE ha sido puesto a prueba frente al cálculo de flujos turbulentos. La baja disipación inherente a los solver de Riemann hace que el esquema sea apto para modelar la turbulencia en un contexto de modelos implícitos LES. La formulación propuesta es capaz de capturar la cascada de energía en el rango de régimen subsónico donde los métodos tradicionales presentan fallos.

Resumo

Desde métodos con malla a métodos sen malla: Unha formulación sen malla xeneralizada baseada en solvers de Riemann para Dinámica de Fluidos Computacional.

Esta tese trata sobre o desenvolvemento de métodos sen malla de alta precisión para a simulación de fluxos compresibles e incompresibles. Os métodos sen malla foron creados para superar as restricións que a conectividade da malla impón sobre os métodos tradicionais. A pesar de ter acadado un éxito relativo nalgunhas aplicacións, a realidade é que os métodos con malla seguen sendo a opción preferente para o cálculo de fluxos que demandan alta precisión. No canto de asumir que os métodos sen malla e con malla son grupos que seguen camiños de desenvolvemento independentes, nesta tese proponse incrementar a precisión dos métodos sen malla tomando como guía algunha das técnicas de máis éxito empregadas na comunidade dos métodos con malla.

O punto de partida para o desenvolvemento inspírase no esquema SPH-ALE proposto por Vila. A flexibilidade do marco de referencia ALE e a introdución dos solvers de Riemann son os elementos esenciais utilizados nesta tese. A alta precisión acádase coa técnica de Mínimos Cadrados Móbiles (MLS). MLS serve para múltiples tarefas na implementación do esquema: acadar alto orde de reconstrución nos estados de Riemann, avaliacións máis precisas dos fluxos viscosos e troco da aproximación limitada tipo kernel por unha aproximación MLS con grado de consistencia polinómica arbitraria. A estabilización do esquema para fluxos compresibles con descontinuidades baséase nunha técnica de estabilización *a posteriori* (MOOD) que introduce unha importante mellora con respecto a os tradicionais limitadores de fluxo *a priori*.

O esquema MLSPH-ALE é a primeira formulación sen malla proposta que emprega a técnica de aproximación MLS con alta consistencia nun marco de referencia ALE. Ademais, o procedemento seguido para obter a forma semi-discreta realiza o seguimento dun termo na fronteira que facilita a implementación das condicións de contorno.

Outra importante contribución relacionase co concepto xeral da formulación MLSPH-ALE proposta. Demostrase que o esquema MLSPH-ALE é unha formulación sen malla global que con certas configuración particulares rende as mesmas formas semi-discretas que outras formulacións publicadas.

O método MLSPH-ALE foi posto a proba fronte o cálculo de fluxos turbulentos. A baixa disipación implícita aportada polo solver de Riemann fai que o esquema sexa apto

para acometer o modelado da turbulencia cos modelos implícitos LES. A formulación proposta captura a cascada de enerxía no rango de réxime subsónico, onde os métodos tradicionais SPH presentan deficiencias.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Introduction

Historically, advances in Fluid Dynamics have promoted great development in society. Infrastructures for water supply and sanitation, transportation and energy production are among the most known flow applications in engineering field. In the industrial sector, Fluid Dynamics is paramount in the design of lubrication systems and fluid power control, for example. But apart from the classical engineering fields, Fluid Dynamics drives the development of modern biomedical devices, like mechanical heart valves, and are the base of models aimed to predict the weather or natural disasters.

Nowadays, it is recognized that Fluid Dynamics advances can be tackled by any of these tracks: experimental, theoretical/analytical and computational. They can be considered as the three branches of the Fluid Dynamics, but interactions between branches are usual.

The interrelation between Fluid Dynamics branches has evolved over time. In a first period of humankind, the only track of fluid knowledge was the experimental approach. The governing equations of inviscid fluid motion by Euler [Euler, 1757] and the inclusion of friction by Navier [Navier, 1823] and Stokes [Stokes, 1845] are the great milestones of the analytical branch. In Anderson [Anderson Jr., 2010], a brief history of the early development of theoretical and experimental parts is provided. The Computational Fluid Dynamics branch came into the picture in the first part of 20th century, and it is common refer to it by its acronym CFD. The growth of CFD has been pushed by the evolution in the computing power and the great advances in numerical methods.

In the current state-of-the-art, the three branches of Fluid Mechanics are receiving great attention by researchers. The most known relation between the theoretical branch and CFD is the use of some known problems with analytical solution to conduct the verification of codes. But nowadays the reciprocal relation also holds, and

the theoretical branch uses CFD simulations to verify hypothesis or to formulate new ones. CFD has gained popularity among technological companies in order to reduce the time-to-market of their products. Hydrodynamic and wind tunnels are essential to provide the experimental performance of a product. However, CFD virtual environments can guide engineers to develop better products in less time and at a lower cost. In Table (1.1), taken from [Basic, 2021], a list of differences for several categories is collected

Category	Physical experiment	Numerical simulation
Results	Flow measurements	Flow prediction
Readings	Equipment-limited number of points	All quantities at any
Price	Relatively expensive	time/space
Time	Slow process	Relatively cheap
Scaling	Smaller models	Relatively quick
Setup	Limited range of operating conditions	Any operating conditions
Repeatable	Mostly	Yes
Safe	Not all	Yes
Error sources	Measurement errors, flow disturbances, etc.	Discretization, numerical method, etc.

Table 1.1. Comparison between numerical simulations and experiments. Adapted from [Basic, 2021]

This thesis deals with the computational branch of Fluid Dynamics. In particular, the main contributions of this work are contained in a set of numerical methods known as meshless methods.

1.2. Computational Fluid Dynamics. To mesh or not to mesh?

There is an extensive list of numerical methods in CFD, and each comes with its own advantages and disadvantages. The existence of these methods is motivated by the wide range of fluid applications that demand a numerical solution. In this section, an outline of numerical methods for CFD is given, with the only purpose of providing the scope of the present work.

The classical methods in Fluid Dynamics are the Finite Difference Method (FDM), the Finite Volume Method (FVM) and the Finite Element Method (FEM). These methods have been the core of the main software engineering packages. Detailed descriptions of the basis of these methods can be found in reference CFD textbooks [Hirsch, 2007; Blazek, 2015; Zienkiewicz et al., 2014]. Although they are called classic, they are continuously evolving to improve accuracy, performance or solving more complicated flow problems. These methods share the common feature that the domain of interest is discretized on a grid/mesh and the governing equations are approximated in each

node, cell or element. Another classical method denoted as Boundary Element Method (BEM) could be included in this category, but BEM only requires the discretization of the domain boundary [Brebbia et al., 1984]. When the geometric domain is discretized into nodes, volumes, or elements, information about the connectivity between these divisions must be known so that systems of equations can be assembled. Mesh generation is a major portion of the work that must be done in order to perform accurate CFD analysis. When moving or deforming regions in the domain appear, the evolution of a topological mesh could deteriorate the accuracy and even cause an abrupt fail of the numerical computation. In the practical use of Computer Aided Engineering (CAE) software, failed simulations are frequently caused by an initial low quality mesh or degenerated mesh. The tips and tricks to create a good mesh are so demanding that meshing task sometimes seems an art.

Solving Navier-Stokes equations using mesh-based methods have been the most usual choice in both scientific computations and industrial applications. Currently, mesh-based second-order schemes are the common standard in CFD simulations. However, the effect of irregular meshes decreases the accuracy of the simulations, since using those grids the effective order of the numerical schemes is between one and two. Thus, in most of engineering flows, a very fine mesh is required to obtain highly accurate solutions, and frequently it is not possible to obtain a grid-independent and reliable solution.

There is clear evidence that current standard methods are unable to deal with a number of important engineering and scientific problems. Following Kroll [Kroll, 2006] the current standard simulation tools for CFD present the following problems:

- High Reynolds number aerodynamic simulations of viscous flows around complex geometries require a high amount of computational power.
- Advanced turbulence models (Large Eddy Simulations (LES), for example) are still only rarely used for industrial applications.

Even though these problems are particularized for the aeronautical industry, they are shared by other disciplines where flow problems need to be addressed. It is reasonable to assume that in the future, numerical methods of high accuracy will have an increasingly important role to play. Thus, a crucial question arises: Given an error considered acceptable, what is the best way to achieve that error in terms of the amount of memory and processing time required? Do we use a low–order method on a fine mesh or a high–order method on a coarse mesh? Available evidence [Schwartzkopff et al., 2002] [Schwartzkopff et al., 2004] suggests that the second option is by far the most efficient way to reduce errors, by orders of magnitude. Thus, great research effort is currently being devoted to the development of highly accurate numerical methods [Kroll, 2006] [Sherwin & Peiró, 2001] [Wang et al., 2013]. Unfortunately, high-order methods are less robust than standard methods. Moreover, curved adapted meshes are required in order to keep the accuracy of the numerical scheme to acceptable levels, and there are no efficient mesh generation technologies to generate these grids.

Therefore, it is known that current state-of-art grid-based high-order methods have the following drawbacks [Sherwin & Peiró, 2001] [Wang et al., 2013] :

- They usually require specialized and efficient implementation, since they are more sophisticated than low-order methods from the mathematical point of view.
- They require much more memory (with implicit schemes).
- Some of them need robust high-order mesh generators that are not readily available.
- They introduce very little numerical dissipation. Therefore, shock-capturing and time integration are more challenging than for low-order methods.

These significant limitations highlight the need to explore other alternatives that avoid the cumbersome task of generating a high quality mesh and circumvent the aforementioned limitations of high-order grid-based methods. Although some meshing packages have automatized the tasks of meshing, a numerical procedure that avoids any meshing has attracted the interest of many researchers.

Meshless methods do not require any geometric data structure with connectivity between entities to provide a numerical solution. The computational domain for a meshless method is simply a cloud of points. The cloud of points is more flexible than a grid and even the generation technique is about an order of magnitude faster [Löhner & Oñate, 1998]. However, meshless methods in Fluid Dynamics have not achieved the same maturity state as mesh-based methods. The first attempt to solve partial differential equations with a meshless framework was attributed to Lucy [Lucy, 1977] and Gingold and Monaghan [Gingold & Monaghan, 1977] who independently conducted simulations in the field of Astrophysics using the Smoothed Particle Hydrodynamics (SPH) method. Although meshless methods circumvent the problems associated with mesh, the alternative path comes with a new set of difficulties. Some of them are the imposition of some type of boundary conditions, more demanding shape functions and more complex stencils. Although SPH is the most widespread meshless method used in CFD there are other meshless methods receiving attention by researchers. In the literature it is common in a first approximation to assign to meshless methods the features of SPH. In state-of-the-art meshless methods review, the distinction between the traditional SPH formulation, corrected versions of SPH and other meshless formulations will be detailed.

Meshless methods are very attractive due to their low preprocessing efforts and high postprocessing capabilities. In addition, Lagrangian meshless methods simulate free-surface flows in a more natural way than mesh-based methods. But apart from these favorable characteristics, meshless methods suffer from other deficiencies. Meshless methods still have not achieved the accuracy and convergence rates of advanced mesh-based methods, and in general the specification of some types of boundary conditions is more cumbersome in meshless methods than in mesh-based methods.

To sum up, this section was started by listing the weak points of mesh-based methods. Then meshless methods were introduced as an alternative that avoids the issues associated with the mesh but this procedure comes with some deficiencies in accuracy and convergence rates. So the question "to mesh or not to mesh" remains inconclusive. Maybe several decades later, one procedure will defeat the other or maybe both procedures will be in use for researchers and engineers. Nowadays, meshless methods can be considered as a promising numerical methodology with respect to the more assented mesh-based ones. Based on that consideration, it seems reasonable to explore the potential of meshless methods looking for an improvement of its accuracy and convergence rates.

1.3. Applications of meshless methods in Fluid Dynamics

Meshless methods have attracted the attention of many researchers to model fluid phenomena that can not be tackled easily by mesh-based methods. In general, meshless methods are selected by researchers and engineers in applications with the presence of interfaces and/or complex geometric domains. In order to be more specific we can cite the following topics in fluid mechanics where meshless methods found widespread use: free surface flows [Monaghan, 1994; Gomez-Gesteira et al., 2010], multiphase flows [Wang et al., 2016], fluid-structure interaction [Liu & Zhang, 2019], sloshing [Colagrossi et al., 2010], floods and tsunamis [Vacondio et al., 2013], lava flows [Bilotta et al., 2015], casting [Cleary et al., 2014] and lubrication [Schnabel et al., 2018],among others.

The previous list is not exhaustive and is biased towards civil and industrial applications. Meshless methods are applied in these types of flows but it does not mean that they overperform mesh-based methods. In fact, mesh-based methods have evolved to incorporate techniques that improve the treatment of sharp interfaces or the interaction with moving objects. For instance, the Arbitrary Lagrangian-Eulerian (ALE) and the Immersed Boundary (IB) are techniques that boost the possibilities of mesh-based methods to deal with complex moving boundaries. Similarly, multiphase techniques like the Phase Field Model (PFM) and the Level-Set Functions (LSF) have enabled mesh-based methods to improve the resolution of interfaces. The selection of the most convenient formulation to solve a specific application is a controversial issue. Accuracy and computational cost are always considered to evaluate numerical methods. However, the time to prepare a model and the availability of a robust software package are also very important factors that have to be considered.

There is no single way to present the increasing importance of meshless methods in Computational Fluid Dynamics. From an academic point of view, the number of publications that use meshless methods has increased notoriously. There are journals dedicated solely to meshless methods. There are also journals that have considered special issues dedicated to the development of meshless methods. The growing activity of the SPHERIC community [1] can also serve as an indicator of the increasing attention

[1] https://www.spheric-sph.org

received by SPH that is considered the first and most widely used meshless method.

The increasing importance of meshless methods can also be sensed by examining the last acquisitions done by Computational Aided Engineering (CAE) software companies. In a short period of time, giant CAE companies with renowned mesh-based methods have acquired companies developing particle based methods. The fact that commercial CAE companies have incorporated meshless methods in their catalog of products is a good indicator that meshless methods have a great potential to provide solutions for demanding industrial problems.

With the advent of meshless methods, the possibilities in the virtual development of products are extended. When a researcher or engineer demands a numerical solution for a problem, incorporating a new suite of tools is always a desirable fact. There are some applications dealing with fluids that can be solved more efficiently with mesh-based methods than with meshless methods. For other set of applications, the meshless methods would be preferred over mesh-based methods.

A very representative application is concerned with the design of tires to improve its performance in wet condition. Aquaplaning in Vehicle Dynamics is a key factor in the design for tire manufacturers. Computational simulations for analyzing aquaplaning have only been conducted by using meshless methods like SPH [Chiron et al., 2019]. Pelton turbines [Marongiu, 2007; Vessaz, 2015], lubrication of gearboxes [Ji et al., 2018], ditching of airplanes/helicopters in the ocean [Oger et al., 2020] are other examples in the energy, mechanical and transport industry that are more suited to be modeled by meshless formulations.

For some demanding problems a coupling strategy between a mesh-based and meshless method could be the most efficient approach [Marrone et al., 2016; Fernandez-Gutierrez et al., 2017]. Meshless methods can also provide a smooth transition between different meshes in a Finite Volume scheme. By taking the best of meshless and mesh-based methods is possible to conceive hybrid methods that are meshless but incorporate some features of FVM. These methods are classified as meshless-FV methods in the literature [Hopkins, 2015].

1.4. State-of-the-art of meshless methods for Fluid Dynamics

The development of meshless methods follows the path of any novel idea in science. After the impact of first publications, some new contributions appear years later that try to extend the method and circumvent some issues detected. Although the main interest in this thesis is concerned with CFD, we remark that important achievements in meshless methods were reached in the study of other fields of knowledge, especially in Solid Mechanics.

Meshless methods for Fluid Dynamics encompass a wide range of methods. In the literature these methods are also designated as meshfree or gridfree. Moreover, since the calculation entities of meshless methods are usually called particles it is common to associate meshless with a particle based concept. This is not an equivalent association

since, for instance, the Lattice Boltzmann Method is a particle based method that it is not a meshless method. Classifications of meshless methods according to different criteria are given by Liu's book [Liu & Liu, 2003] and Gu's paper [Gu, 2005]. A visual map containing an extensive list of meshless methods is provided in Douillet [Douillet-Grellier, 2019]. In the family of Particle/Meshless Methods we can made a distinction between methods aimed to solve general PDEs in the continuum scale and methods aimed to solve discrete systems. Molecular Dynamics, Discrete Element Method and the Lattice Boltzmann Method are representative particle methods aimed to solve discrete systems. However, this thesis is concerned with meshless methods aimed to solve the continuum Fluid Dynamics equations. The first and most renowned meshless method of this type is the Smoothed Particle Hydrodynamics (SPH). Therefore, the review given in this section outlines the developments of SPH and some other meshless formulations that were conceived to improve the performance of SPH methods.

As already mentioned, the origin of meshless methods is usually dated with the first papers of SPH published independently by [Lucy, 1977] and [Gingold & Monaghan, 1977] to solve gas dynamics problems in Astrophysics. SPH is a Lagrangian meshless technique that discretizes the continuous domain in a set of moving particles. The continuum is represented by convolution of particle properties with a regularizing function that is common designated as kernel. SPH enjoys some very desirable properties: conservation of mass, momentum and energy at machine precision. The Lagrangian motion of particles makes SPH a suitable method for solving advection-dominated dynamics. SPH method did not attract the interest of researchers in other fields until the beginning of the 1990s, when the method was applied in fields such as impact penetration in solids [Libersky et al., 1993; Randles & Libersky, 1996]. With the application of SPH to Solid Mechanics the method was receiving much more attention by the scientific community and some deficiencies of the method were detected. The lack of consistency for even a constant field and the low-order accuracy are the main deficiencies that contributed to discard SPH method when compared with mesh-based methods.

A numerical approximation is said to be p-order consistent when it is able to reproduce exactly polynomial fields of order less than or equal to p. The definition is valid for either the approximation of a field function or the approximation for its gradient. The original kernel approximation used by SPH is unable to provide an exact approximation for a uniform constant field, that using the consistency definition is expressed by saying that kernel approximation for a field lacks of zeroth-order consistency. Similarly, the original kernel approximation for the gradient is unable to provide and exact approximation for the gradient of a constant uniform field and therefore the kernel approximation for the gradient lacks of zeroth-order consistency.

Original SPH formulations suffer from low accuracy. The kernel approximation is based on the identity property of the convolution of a function with the Dirac function. The kernel approximation can be decomposed in two sequential approximations. In the first step, the Dirac function is replaced by a regularizing kernel function and in

a second step the integral is approximated by a summation over a set of neighbors. Although the first approximation is second-order accurate the discretization involved in the second step degrades the accuracy and therefore the kernel accuracy order is lower than 2.

In order to remedy the issues of SPH, it is possible to identify three strategies in the proposals made by researchers in 1990s decade.

A first research line, close to the original SPH procedure, is focused on improving the kernel approximation. Liu et al. [Liu et al., 1995] suggested introducing adjusting factors in the traditional kernel function that allows for exact reproduction of constant, linear or higher order fields. Liu incorporated this enhanced kernel approximation in the SPH formalism resulting in the Reproducing Kernel Particle Method (RKPM). Bonet and Kulasegaram [Bonet & Kulasegaram, 2000] adopted this procedure for reproducing linear fields exactly, resulting in the Corrected Smoothed Particle Hydrodynamics (CSPH). Frontiere et al. [Frontiere et al., 2017] presented the Conservative Reproducing Kernel SPH (CRKSPH) scheme. CRKSPH can be considered as a reformulation of RKPM that assures rigorously the conservation of mass, linear momentum and energy. Randles and Libersky [Randles & Libersky, 1996] introduced a renormalization technique to produce a consistent approximation for the gradient of linear fields. Another more powerful technique to restore the particle consistency consist in replacing the kernel approximation by the Moving Least Square (MLS) approximation. The use of MLS for the approximation in SPH was first proposed by Dilts [Dilts, 1999, 2000].

A second research line considers that in order to circumvent the issues of SPH is better to abandon the kernel approximation and adopt the same Galerkin technique used by FEM to develop meshfree methods based on integral weak forms. Meshfree weak form methods require a background mesh to perform the integration of the weak form. Despite the use of this auxiliary background mesh for the integration, these methods are still considered meshless. These methods have found more applications in Solid Mechanics than in Fluid Mechanics. Examples of meshfree weak form methods include: Diffuse Element Method (DEM) [Nayroles et al., 1992], Element Free Galerkin (EFG) [Belytschko et al., 1994] and Meshless Local Petrov Galerkin (MLPG) [Atluri & Zhu, 1998]. DEM is the first meshless method that incorporate the Moving Least Squares approximation technique in the solution of PDEs. EFG can be considered an improved version of DEM that uses complete expressions for the gradient of shape functions and a better evaluation of the integral form. MLPG is a very versatile meshless method based on a local weak form over subdomains of particles that overlap. Both DEM and EFG require a global background mesh to perform the integration, meanwhile MLPG only requires a local background mesh.

The third path to develop meshfree methods adopts the collocation technique (using the strong form). Examples of these methods include: the Generalized Finite Difference Method (GFDM) [Cheng and Liu (2002); Finite Point Method (FPM) [Oñate et al., 1996a], Meshless Collocation Method [Kansa, 1990]. In most meshless methods based on the strong form the points are stationary and therefore they resemble more the

FDM than the SPH method.

Apart from being considered as one of the fathers of the method, Monaghan has written important reviews of SPH method [Monaghan, 1992, 2005] and has conceived a set of innovations to the original SPH formulation. A non exhaustive list reads: artificial viscosity to model shock waves [Monaghan & Gingold, 1983], incorporation of Riemann solvers [Monaghan, 1997], SPH modeling of free-surface flows [Monaghan, 1994], formal derivation of the method using the Lagrange equations [Monaghan, 2005] and a new technique to update the motion of particles [Monaghan, 1989]. In addition, Morris, a disciple of Monaghan, has introduced an improved second-order derivative formulation to treat viscosity and study low Reynolds flows [Morris et al., 1997]. Monaghan's contributions serve as inspiration for many other researchers in the development of SPH. The extension of SPH method to model free-surface flow is considered an important breakthrough because it encouraged researchers to model incompressible flows with meshless formulations. Thereafter, an important development of SPH and related meshless methods are driven by researchers and engineers working in the oceanic and coastal flows. A compressible flow problem in Astrophysics and an incompressible flow in coastal engineering pose very different requirements to numerical schemes. As a consequence, SPH methods for compressible and incompressible flows evolve improving different features.

For compressible flows the simulation of shock waves and contact discontinuities is of vital importance. Inutsuka [Inutsuka, 2002] reformulates the SPH with Riemann solvers in the so-called Godunov–SPH schemes(GSPH). GSPH methods based on the Riemann solver can handle severe problems with strong shocks. In addition, GSPH methods does not require to adjust the constants of the standard SPH with artificial viscosity for stabilization.

Regarding incompressible flows, the first SPH attempts to model these flows adopted a weakly compressible approach, as suggested by [Monaghan, 1994]. SPH methods adopting the weakly compressible approach are designated with the acronym WCSPH. The weakly compressible approach considers that the fluid density is not constant but rather follow a barotropic law. By using a barotropic Equation Of State (EOS) the pressure is related with the density and therefore it can be solved with the algorithm of a density-based solver. Unfortunately, WCSPH results in noise pressure fields and nonphysical oscillations. To circumvent these deficiencies different paths can be followed. Some authors propose to correct WCSPH by introducing additional dissipation terms or introducing Riemann solvers. Marrone et al. [Marrone et al., 2011] introduced a diffusive term in the continuity scheme and use the name δ-SPH to refer to this variant of WCSPH. Parshikov and Medin [Parshikov & Medin, 2002] introduced Riemann solvers to stabilize the WCSPH. A recent paper published by Green et al. [Green et al., 2019] shows that δ-SPH is equivalent to the adoption of a Riemann solver in the continuity equation. There are other SPH methods that abandon the weakly compressible approach and propose to solve the incompressible flow equations. SPH methods that solve the incompressible flow equations are designated as ISPH and they

need to solve a Poisson equation for the pressure enforcing the null divergence of velocity. Cummins and Rudman [Cummins & Rudman, 1999] were the first authors that use a projection method in an SPH formalism. Koshizuka et al. [Koshizuka & Oka, 1996] presented the Moving Particle Semi-Implicit (MPSI) a meshless method that solve the incompressible flow equations but using an approximation that differs from the kernel approximation used by SPH. A comprehensive book about the MPSI method has been published by Koshizuka et al. [Koshizuka et al., 2018].

A great landmark in the development of meshless methods was the work conducted by Vila [Vila, 1999] and Ben Moussa and Vila [Ben Moussa & Vila, 2000]. Vila incorporated the ALE feature for meshless methods and provided a systematic way to obtain the meshless solution of the weak form associated with a conservation law. In the literature the SPH formalism used by Vila is usually referred as SPH-ALE method. The resulting semi-discrete system obtained by Vila results in an SPH formulation that is very close to the Finite Volume formalism based on Riemann solvers. Since interaction between particles are interpreted as Riemann fluxes, the local conservation property is assured and enough diffusion is provided to stabilize the method. In this formulation the artificial viscosity is replaced by the intrinsic numerical viscosity of the Riemann solver. The first formulation obtained by Vila formally resembles a first order Godunov method with constant piecewise reconstruction. In the mesh-based community is a well documented fact that Godunov method leads to very dissipative schemes. A great advantage in the formulation proposed for Vila is the formal similarity with the Finite Volume Method. We can take advantage of this analogy to improve the accuracy and features of the meshless technique by using techniques developed in FVM. Based on the similarity form of SPH-ALE with FVM Vila proposed in [Vila, 1999] a high order extension of SPH-ALE following the MUSCL technique introduced by van Leer [van Leer, 1979]. The work conducted by Vila inspired the works of other authors to develop meshless method with close similarity with the FVM. Among these works we mention the Finite Volume Particle Method (FVPM) by Hietel [Hietel et al., 2000] and the Gizmo-MFV by Hopkins [Hopkins, 2015]. The SPH-ALE method was extended to solve incompressible flow problems by invoking the weakly compressible approach. Several improvements were incorporated in the WCSPH-ALE method: Marongiu [Marongiu, 2007] implemented boundary conditions based on partial Riemann solvers, Oger [Oger et al., 2016] incorporated a Particle Shifting Technique in a WCSPH-ALE formulation that can deal with free-surface flows. Colle and Vila [Collé et al., 2019] presented the γ-SPH-ALE scheme for barotropic flows, that is based on the combination of the SPH-ALE with a Finite Volume low-Mach scheme.

The Finite Volume Particle Method (FVPM) introduced by Hietel et al. [Hietel et al., 2000] is a meshless method that was conceived as Finite Volume Method implemented in the data structure of a particle meshless method. Particles in FVPM move with a transport velocity field and therefore enjoys the advantages of an ALE framework. The main novelty of FVPM is to associated a particle shape function that automatically satisfies the partition of unity property to each particle location. Keck

and Hietel incorporate a projection technique in FVPM for incompressible flow [Keck & Hietel, 2005]. Quinlan and coworkers have conducted a very active research work on the FVPM on the last decade. Some of their main contributions are the extensions to viscous flow [Nestor et al., 2009] and free-surface flows [Quinlan, 2018]. In the extension to viscous flow, a particle shifting technique accommodated in an ALE framework is included for the first time. Although SPH-ALE and FVPM were derived following different concepts both meshless methods result in semi-discrete equations very similar to the ones obtained with a FVM. Based on that, extensions developed for SPH-ALE or FVPM can be easily transferred from one formulation to the other.

Meshless-fv methods like SPH-ALE and FVPM in their most basic configurations, resemble to a Godunov method in FVM. Therefore, SPH-ALE and FVPM can handle shocks without requiring any artificial viscosity term for stabilization. However, Godunov method is excessively diffusive and we need to incorporate extensions in the basic configurations of SPH-ALE and FVPM to increase the accuracy. The accuracy of the method improves when higher-order reconstructions of the Riemann states are performed. Different techniques to increase the order of the reconstruction of the Riemann states in different meshless methods are used in the literature. Hopkins [Hopkins, 2015] adopts a MUSCL-Hancock procedure with limiter. Avesani et al. [Avesani et al., 2014] introduced a polynomial Weighted Essentially Non-Oscillatory (WENO) reconstruction using MLS. Nogueira et al.[Nogueira et al., 2016a] also adopt MLS to reconstruct Riemann states but the stabilization is achieved with Multi-dimensional Optimal Order Detection (MOOD) technique.

Despite the fact that meshless and mesh-based methods have advantages and drawbacks derived from the different form to accomplish the discretization of the partial differential equations, there are some physical phenomena (like turbulence) that are very demanding computationally, for both types of approaches. The research on the numerical simulation of turbulent flows has been mainly addressed by using mesh-based methods. Simulation of turbulence using meshfree methods has received less attention, and many studies deal with complex applications without providing details of their development, diffusion rates or details about the evolution of the energy spectrum. We refer the reader to[Monaghan, 2002, 2011; Mayrhofer et al., 2015; Hu & Adams, 2015; Di Mascio et al., 2017] for some examples.

In the last decade it is possible to find some reviews of meshless methods focused on particular applications. For instance, Gotoh and Khayyer [Gotoh & Khayyer, 2018] provide a review on particle methods with applications in coastal and ocean engineering. Violeau and Rogers [Violeau & Rogers, 2016] depict about recent trends in SPH methods focused on the modeling of free surface flows.

In order to guide researchers and engineers to improve SPH methods, the SPHERIC organization [2] was conceived in 2005. In a recent paper [Vacondio et al., 2020], SPHERIC members encourage researchers to focus the attention on topics where SPH

[2]SPHERIC is the acronym of SPH European Research Interest Community

suffer drawbacks in comparison with mesh-based methods. These are the current (2021) SPH Grand Challenges:

- GC1: Convergence, consistency and stability
- GC2: Boundary Conditions
- GC3: Adaptivity
- GC4: Coupling to other methods
- GC5: Applicability to industry

This work contributes to GC1, GC2 and GC4.

1.5. Research lines in the Group of Numerical Methods in Engineering (GMNI)

Along the last decades, the GMNI[3] has developed numerical methods to solve applications in several engineering fields. Concerning applications dealing with fluids, contributions using multiple methods were accomplished. Based on that experience, synergies between meshless and mesh-based methods were invoked to develop high-accurate numerical methods in a wide range of flow conditions. Among others, Cueto-Felgueroso et al. [Cueto-Felgueroso et al., 2004] developed a meshless formulation derived on a Discrete-Galerkin approach. Nogueira et al. [Nogueira et al., 2010] presented a shock-capturing technique for unstructured grids based on MLS technique. Ramírez et al. [Ramírez et al., 2014] achieved high order accuracy in the solution of incompressible Navier–Stokes by incorporating the MLS approximation. Ramírez et al. [Ramírez et al., 2017] demonstrate that MLS technique can preserve the accuracy in moving sliding interfaces. In a more recent work, Fernández-Fidalgo et al. [Fernández-Fidalgo et al., 2020] combined a high order FDM with MLS to prescribe solid wall boundary conditions in curved domains.

The present work continues the research on these lines, but focusing on meshless methods.

1.6. Motivation of this work

In the current state-of-the-art, there is agreement in the grand challenges that must overcome meshless methods to increase its reliability. Increasing the accuracy and convergence rates of meshless methods is considered one of the main challenges of these kind of schemes. GMNI has introduced some improvements in mesh-based methods by using techniques originated in meshless techniques. In this work the transfer between

[3] https://caminos.udc.es/gmni

methods is in the reverse direction. Improvements in the accuracy of meshless methods are suggested based on the guidance provided by mesh-based methods.

The present thesis is included in the GMNI research line of boosting meshless methods for Fluid Dynamics. In particular, this work is focused on the development of a high accuracy meshless scheme to solve Euler and Navier–Stokes. The scope of this work is only concerned with the development of meshless formulations itself, without having in mind one specific application.

1.7. Objectives

The main objective of this thesis is to develop a new high-accuracy meshless numerical method that can overcome the drawbacks of current CFD state-of-the-art grid based high-order methods, for applications in turbulent flows and complex problems. The new algorithms will enable the development of new and more efficient simulation tools for CFD computations.

In order to achieve the main objective of the project, the following specific objectives will be pursued:

- To improve the accuracy of kernel approximation in SPH-like methods: To do that, we propose to develop a new formulation based on Moving Least Squares for the integration of the equations instead of using kernel approximations
- To develop a high-accuracy meshless method for weakly compressible flows.
- Integrate the boundary condition treatment in the formulation.
- Unify existing meshless formulations into a more generalized framework.
- To develop a family of meshless methods which are able to compute accurately truly turbulent flows.

1.8. Document structure

The document is organized into 7 chapters, which will be briefly described in the following.

Chapter 1 provides a brief state-of-the-art of meshless numerical methods and a presentation of the motivation and objectives of this PhD.

In Chapter 2 the governing equations are presented in several forms. Then the fundamentals of the SPH are given and the derivation of the weak formulation of the SPH-ALE method is detailed.

Chapter 3 deals with the development and verification of a high accuracy SPH-ALE method for weakly compressible flow. Some novelty aspects incorporated in this formulation are the use of MLS to increase the order of reconstruction of the Riemann

states, a weakly-compressible version of the *a posteriori* stabilization and the treatment of viscous terms with MLS.

In Chapter 4 we present some modifications to the method proposed in Chapter 3 in order to improve the accuracy and to increase the range of applicability of the methods. Instead of using the kernel or normalized kernel approximations this method lends MLS technique the task of integrating the weak form and approximating the solution. The method is designated as MLSPH-ALE because the MLS technique is integrated in the SPH-ALE method and not used as a complement to incorporate additional features of the method. A detailed derivation of the method is included. *A posteriori* stabilization is proposed to extend the formulation to compressible flows with the presence of discontinuities. The method assures local conservation and enable to prescribe boundary conditions with a boundary integral term.

In Chapter 5 it is shown that the proposed MLSPH-ALE is a general formulation that includes most of the existing meshless methods. Although the derivation procedures are different the resulting semi-discrete forms adopt a common form. MLSPH-ALE provides a very general framework for meshless-fv formulations. By adopting particular settings in MLSPH-ALE method, it is possible to arrive at the semi-discrete form obtained with other formulations published in the literature.

Chapter 6 tests the MLSPH-ALE method to compute turbulent flows. By conducting some turbulent cases is proved that MLSPH-ALE method is a numerical scheme with the potential to be used in Large Eddy Simulations (LES).

Chapter 7 draws the main conclusions of this work and gives future research lines.

Chapter 2

The SPH-ALE method

2.1. Introduction

The aim of this chapter is to provide a comprehensive historical review of the SPH-ALE method from the pioneering work of Vila [Vila, 1999] to the most recent meshless formulations.

The starting point for the chapter is the presentation of the governing equations in a form that facilitates the derivation of the method. Then, the basic principles of the SPH approximation are exposed. After that, the derivation of the SPH-ALE method is accomplished. Given the great importance of Riemann problems in the SPH-ALE method, an overview of the utilization of approximate Riemann solvers is given. Section 2.6 presents the different strategies to implement boundary conditions. An outline of different published works aimed to increase the accuracy of SPH-ALE is delayed to Chapter 3.

2.2. Governing Equations

In this section the equations of motion for fluids are presented assuming a continuum model for the fluid. To describe the motion of the fluid some definitions widely used in Continuum Mechanics are adopted.

We explain the main definitions by referring to Figure 2.1. We designate by $\widehat{\Omega}$ a continuous material body and denote by $\widehat{P}$ a material particle. The material body moves in the Euclidean space $\mathcal{E}$ and occupies different regions at different times. We pick a point O in $\mathbb{R}^3$ as the origin of a fixed Cartesian Coordinate System. The motion of the body during a time interval from an initial time t_0 to the current time t can be described as a sequence of configurations in space. There exist a correspondence between any material particle $\widehat{P}$ in the material body $\widehat{\Omega}$ and a point in space for any configuration. The initial configuration of material body $\widehat{\Omega}$ at the initial time $t = t_0$

is designated as $\widehat{\Omega}_0$ and the configuration at the current time t is designated as $\widehat{\Omega}_t$. In the initial configuration $\widehat{\Omega}_0$, the point with vector position $\boldsymbol{X}$ is the location in the Euclidean space that occupies the particle $\widehat{P}$ at the initial time t_0. Similarly, in the current configuration $\widehat{\Omega}_t$, the point with vector position $\boldsymbol{x}$ is the location in the Euclidean space that occupies the particle $\widehat{P}$ at a current time t. The successive positions occupied by particle $\widehat{P}$ from the initial position $\boldsymbol{X}$ to the current position at $\boldsymbol{x}$ describes the trajectory of particle $\widehat{P}$ in the interval of time from t_0 to t. We denote by $\widehat{\varphi}_t$ the mapping between the initial configuration $\widehat{\Omega}_0$ and the current configuration $\widehat{\Omega}_t$ at time t. Scovazzi and Hughes [Scovazzi & Hughes, 2007] refer to $\widehat{\varphi}_t$ as the Lagrangian-to-Eulerian map.

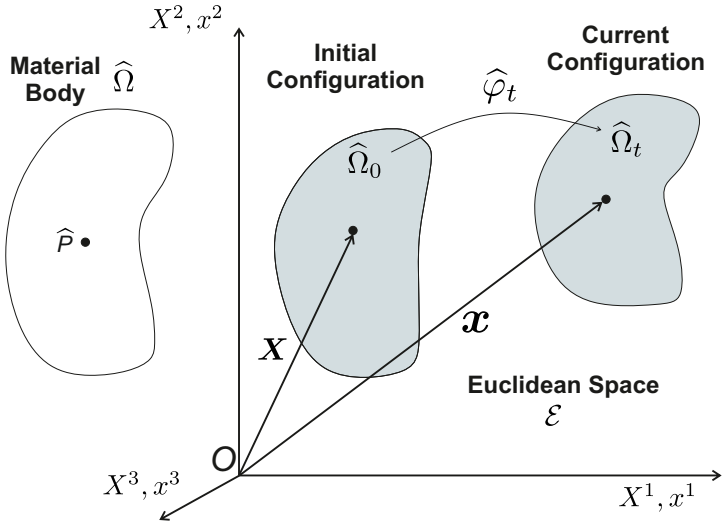


Figure 2.1. Motion description of a material body $\widehat{\Omega}$ in Euclidean Space $\mathcal{E}$ with initial configuration $\widehat{\Omega}_0$ and current configuration $\widehat{\Omega}_t$. A material particle $\widehat{P}$ occupies position $\boldsymbol{X}$ at initial time and position $\boldsymbol{x}$ at current time t.

In Figure 2.1 we depicted the initial configuration $\widehat{\Omega}_0$ and the current configuration $\widehat{\Omega}_t$ of a material body $\widehat{\Omega}$. In Figure 2.2 we represent the time evolution of the configurations. This space-time representation is useful to remark some details in the description of the fluid motion. For the sake of clarity, configurations are assumed to be contained in a two dimensional plane. On the left, we plot the space-time domain of the Lagrangian framework with the material coordinates $\boldsymbol{X}$. On the right we plot the space-time of the material body with spatial coordinates $\boldsymbol{x}$. The base of the space-time element at time $t = t_0$ are coincident but along the deformation/motion the spatial coordinates $\boldsymbol{x}$ occupied by the material volume evolve with time.

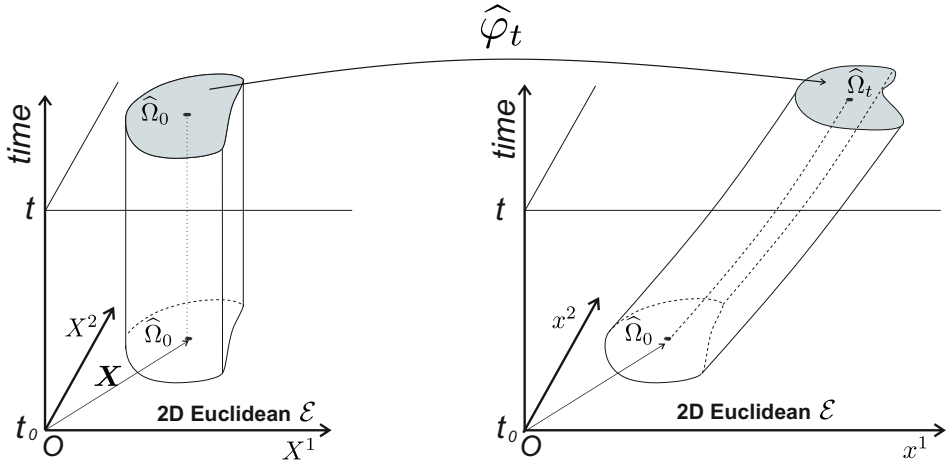


Figure 2.2. Description of fluid motion of a material body $\widehat{\Omega}$ in space-time configuration. On the left: Lagrangian framework with material coordinates $\boldsymbol{X}$. On the right: Eulerian framework with spatial coordinates $\boldsymbol{x}$.

The dynamical laws of motion are stated for a material body as follows:

1. Conservation of mass (continuity): The mass of a material volume is constant.
2. Balance of linear momentum (Newton's second law): The rate of change of a material body momentum is equal to the sum of the surface forces (due to pressure and viscous stresses) and body forces (such as gravity) acting upon it.
3. Balance of energy (first law of thermodynamics): The rate of change of the material body energy (internal plus kinetic) is equal to the rate at which forces do work upon it plus the rate at which heat is transferred to it.

The physical laws apply directly to material bodies. In order to impose the physical laws to arbitrary regions in space, balance equations must be formulated. These regions in space are denoted as control volume in Fluid Mechanics and attached to them the computational domain in CFD studies are defined. To describe the motion of an arbitrary control volume we can adopt the same ideas used to describe the motion of a material body. Figure 2.3 describes the motion of an arbitrary control volume $\Omega(t)$. Somehow, the control volume can be interpreted as a pseudo material body that is comprised of fictitious material particles. The initial configuration of the control volume $\Omega(t)$ is denoted as Ω_0 and the current configuration as Ω_t. A fictitious particle P occupies position $\boldsymbol{\chi}$ at initial time t_0 and position $\boldsymbol{x}$ at current time t. We denote by φ_t the mapping between the initial configuration Ω_0 and the current configuration Ω_t at time t of the control volume $\Omega(t)$. In the space-time representation, φ_t is the Referential-to-Eulerian map since it defines the transit from the Lagrangian framework with referential coordinates $\boldsymbol{\chi}$ to the Eulerian framework with spatial coordinates $\boldsymbol{x}$.

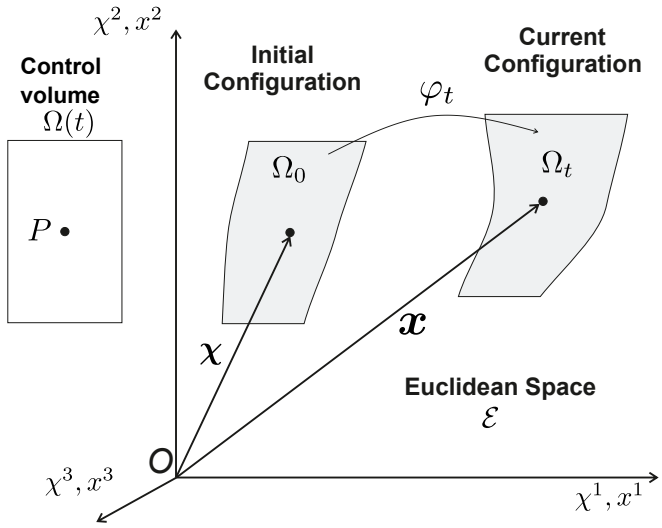


Figure 2.3. Motion description of a general control volume Ω in Euclidean Space $\mathcal{E}$ with initial configuration Ω_0 and current configuration Ω_t. A computational particle P occupies position $\boldsymbol{\chi}$ at initial time and position x at current time t.

2.2.1. Governing Equations in Eulerian form

Governing equations in Eulerian form are obtained by imposing the balance equations to a fixed region in space. The resulting set of partial differential equations in conservative form is given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u^{\alpha}}{\partial x^{\alpha}} = 0 \tag{2.1}$$

$$\frac{\partial \rho u^{\beta}}{\partial t} + \frac{\partial \left(\rho u^{\beta} u^{\alpha}\right)}{\partial x^{\alpha}} = \frac{\partial \tau^{\alpha\beta}}{\partial x^{\alpha}} - \frac{\partial p}{\partial x^{\beta}} + \rho f^{\beta} \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho H u^{\alpha}\right)}{\partial x^{\alpha}} = \frac{\partial}{\partial x^{\alpha}}\left(k \frac{\partial T}{\partial x^{\alpha}}\right) + \frac{\partial \left(\tau^{\alpha\beta} u^{\beta}\right)}{\partial x^{\alpha}} + \rho f^{\beta} u^{\beta} + \dot{q}_h \tag{2.3}$$

where Greek superscripts with summation convention are used to refer spatial directions in a Cartesian Coordinate System with d dimensions. Greek superscripts (α, β) range from 1 to d and are associated with the Cartesian directions in the usual form $1 \equiv x, 2 \equiv y, 3 \equiv z$. The vector fluid velocity is given by $\boldsymbol{u} = (u^1, .., u^d)^T$. Density and pressure are designed by ρ and p respectively. Total energy is denoted by E and its relation with the internal energy e and the kinetic energy reads $E = e + \frac{1}{2}(u^{\alpha\alpha})$. Total enthalpy definition is used to identify $H = E + p/\rho$. The external force component per unit mass in momentum equation are expressed by f^{α} and $\dot{q}_h$ represents a volumetric heat source. For the diffusive terms $\tau^{\alpha\beta}$ denotes the viscous tensor component and thermal conduction flux component is given in terms of thermal conductivity k and

temperature gradient as $q^\alpha = -k\frac{\partial T}{\partial x^\alpha}$. Assuming a Newtonian fluid, the viscous tensor is defined as

$$\tau^{\alpha\beta} = \mu\left(\frac{\partial u^\beta}{\partial x^\alpha} + \frac{\partial u^\alpha}{\partial x^\beta}\right) + \left(\mu_v - \frac{2}{3}\mu\right)(\nabla\cdot\boldsymbol{u})\,\delta^{\alpha\beta} \tag{2.4}$$

where μ and μ_v are the dynamic and volumetric viscosity of the fluid respectively.

Eqs. (2.1), (2.2) and (2.3) can be re-written in a flux vector form by

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_E^\alpha}{\partial x^\alpha} = \frac{\partial \boldsymbol{D}^\alpha}{\partial x^\alpha} + \boldsymbol{S} \tag{2.5}$$

$$\begin{aligned} \boldsymbol{U} &= \begin{pmatrix} \rho \\ \rho u^\beta \\ \rho E \end{pmatrix}, \quad \boldsymbol{F}_E^\alpha = \begin{pmatrix} \rho u^\alpha \\ \rho u^\beta u^\alpha + p\delta^{\beta\alpha} \\ \rho H u^\alpha \end{pmatrix} \\ \boldsymbol{D}^\alpha &= \begin{pmatrix} 0 \\ \tau^{\alpha\beta} \\ k\frac{\partial T}{\partial x^\alpha} + \tau^{\alpha\beta}u^\beta \end{pmatrix}, \quad \boldsymbol{S} = \begin{pmatrix} 0 \\ \rho f^\beta \\ \rho f^\beta u^\beta + \dot{q}_h \end{pmatrix} \end{aligned} \tag{2.6}$$

$$\underbrace{\begin{pmatrix} \rho \\ \rho u^\beta \\ \rho E \end{pmatrix}}_{\boldsymbol{U}} \underbrace{\begin{pmatrix} \rho u^\alpha \\ \rho u^\beta u^\alpha + p\delta^{\beta\alpha} \\ \rho H u^\alpha \end{pmatrix}}_{\boldsymbol{F}_E^\alpha} \underbrace{\begin{pmatrix} 0 \\ \tau^{\alpha\beta} \\ k\frac{\partial T}{\partial x^\alpha} + \tau^{\alpha\beta}u^\beta \end{pmatrix}}_{\boldsymbol{D}^\alpha} \underbrace{\begin{pmatrix} 0 \\ \rho f^\beta \\ \rho f^\beta u^\beta + \dot{q}_h \end{pmatrix}}_{\boldsymbol{S}} \tag{2.7}$$

where $\boldsymbol{U}$ is the vector of conservative variables, $\boldsymbol{F}_E^\alpha$ is the Eulerian flux in direction α of the advective and pressure terms, $\boldsymbol{D}^\alpha$ is the viscous flux in direction α and $\boldsymbol{S}$ is the vector of source terms.

2.2.2. Governing Equations in ALE form

In order to provide compact expressions for the Arbitrary Lagrangian-Eulerian (ALE) framework and following the usual notation in literature, the transport operator is introduced here. Let's denote with $L_{\boldsymbol{w}}(\boldsymbol{U})$ the transport operator with a regular velocity field $\boldsymbol{w}$ that acts on a vector of variables $\boldsymbol{U}$ according to

$$L_{\boldsymbol{w}}(\boldsymbol{U}) \equiv \frac{\partial}{\partial t}\boldsymbol{U} + \nabla\cdot(\boldsymbol{w}\otimes\boldsymbol{U}) \tag{2.8}$$

In a Cartesian Coordinate System with d dimensions the transport operator reads as

$$L_{\boldsymbol{w}}(\boldsymbol{U}) = \frac{\partial \boldsymbol{U}}{\partial t} + \sum_{\alpha=1,d}\frac{\partial}{\partial x^\alpha}\left(w^\alpha \boldsymbol{U}\right) = \frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial}{\partial x^\alpha}\left(w^\alpha \boldsymbol{U}\right) \tag{2.9}$$

where the second equality relation relies on the implicit summation convention over repeated index. The transport operator $L_{\boldsymbol{w}}(\boldsymbol{U})$ is a differential operator whose physical

interpretation is linked to the Generalized Reynolds Transport Theorem, applied to a arbitrary control volume whose boundary is moving with velocity $\boldsymbol{w}$ (see Appendix B).

In order to obtain the ALE form, it is very important to understand the differences between a material volume and a control volume. The material volume is an enclosed region in a Euclidean space composed of the same material particles. The kinematics of a material volume is governed by the motion of their material particles with the physical velocity $\boldsymbol{u}$. A control volume is an enclosed region in Euclidean space whose kinematics must be prescribed with a transport velocity $\boldsymbol{w}$ when defined. Both material volume and control volume move in space and the kinematics of its motion can be described by using either a Lagrangian or Eulerian approach. In order to employ the same terminology for the control volume and for the material volume, it can be useful to interpret the control volume as a fictitious material volume composed of fictitious material particles.

Let $\Omega(t)$ denote an arbitrary control volume that at initial time t_0 occupies the region in the spatial domain Ω_0, and in the current time t occupies the region Ω_t. Let us assume that the initial position of a fictitious material particle is given by $\boldsymbol{\chi}$ and its position at current time is given by $\boldsymbol{x}$. Assuming that fictitious particles belong to $\Omega(t)$ and move with velocity $\boldsymbol{w}$, their fictitious material derivative is given by $\frac{\partial \boldsymbol{U}}{\partial t}\big|_{\boldsymbol{\chi}} = \frac{\partial \boldsymbol{U}}{\partial t} + w^\alpha \frac{\partial \boldsymbol{U}}{\partial x^\alpha}$.

The Generalized Reynolds Transport Theorem states that

$$\frac{\partial}{\partial t}\bigg|_{\boldsymbol{\chi}} \int_{\Omega(t)} \boldsymbol{U}\, dx^d = \int_{\Omega_t} \left(\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial (w^\alpha U)}{\partial x^\alpha} \right) dx^d = \int_{\Omega_t} L_{\boldsymbol{w}}(\boldsymbol{U})\, dx^d \tag{2.10}$$

Note that the Eulerian and Lagrangian version are particular cases that correspond to $\boldsymbol{w} = \boldsymbol{0}$ and $\boldsymbol{w} = \boldsymbol{u}$ respectively. For a material volume $\widehat{\Omega}(t)$, with material coordinates X, the velocity of the boundary $\partial\widehat{\Omega}(t)$ is coincident with the physical velocity ($\boldsymbol{w} = \boldsymbol{u}$) and thus the Reynolds Transport Theorem in the usual form is recovered

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} \boldsymbol{U}\, dx^d = \frac{\partial}{\partial t}\bigg|_{\boldsymbol{X}} \int_{\widehat{\Omega}(t)} \boldsymbol{U} dx^d = \int_{\widehat{\Omega}_t} \left(\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial (u^\alpha \boldsymbol{U})}{\partial x^\alpha} \right) dx^d = \int_{\widehat{\Omega}_t} L_{\boldsymbol{u}}(\boldsymbol{U}) dx^d \tag{2.11}$$

Note that in Eq. (2.11) the common nomenclature used in Fluid Mechanics for the material derivative $\frac{D}{Dt}$ is used with the same meaning than $\frac{\partial}{\partial t}\big|_X$.

In order to show some properties of the transport operator let us assume that $\rho\eta$ (η is any scalar variable) is a general component of the vector of conservative variables $\boldsymbol{U}$. Therefore, any conservative variable in Eq. (2.10) can be rewritten as

$$\frac{\partial}{\partial t}\bigg|_{\boldsymbol{\chi}} \int_{\Omega(t)} \rho\eta dx^d = \int_{\Omega_t} \left(\frac{\partial \rho\eta}{\partial t} + \frac{\partial (w^\alpha \rho\eta)}{\partial x^\alpha} \right) dx^d = \int_{\Omega_t} L_{\boldsymbol{w}}(\rho\eta) dx^d \tag{2.12}$$

Expanding the partial derivatives of the products gives the following relation

$$L_{\boldsymbol{w}}(\rho\eta) = \frac{\partial \rho\eta}{\partial t} + \frac{\partial (w^\alpha \rho\eta)}{\partial x^\alpha} = \rho\left(\frac{\partial \eta}{\partial t} + w^\alpha \frac{\partial \eta}{\partial x^\alpha}\right) + \eta\left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho w^\alpha)}{\partial x^\alpha}\right) = \rho \frac{\partial \eta}{\partial t}\bigg|_{\boldsymbol{\chi}} + \eta L_{\boldsymbol{w}}(\rho) \tag{2.13}$$

showing that transport operator does not satisfy the usual derivative property of the product of two functions. If the general transport velocity $\boldsymbol{w}$ is set to the Lagrangian fluid velocity $\boldsymbol{u}$, Eq. (2.13) results in

$$L_{\boldsymbol{u}}(\rho\eta) = \frac{\partial \rho\eta}{\partial t} + \frac{\partial (u^\alpha \rho\eta)}{\partial x^\alpha} = \rho\left(\frac{\partial \eta}{\partial t} + u^\alpha \frac{\partial \eta}{\partial x^\alpha}\right) + \eta\underbrace{\left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u^\alpha)}{\partial x^\alpha}\right)}_{=0} = \rho \frac{\partial \eta}{\partial t}\bigg|_{\boldsymbol{X}} = \rho \frac{D\eta}{Dt} \tag{2.14}$$

In Eq. (2.14) continuity equation is used $\left(\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u^\alpha)}{\partial x^\alpha} = 0 = L_{\boldsymbol{u}}(\rho)\right)$. It is important to remark the distinction between the Lagrangian transport operator acting on density $L_u(\rho)$ and the material derivative of density $\frac{D\rho}{Dt}$. Mass conservation is expressed with the Lagrangian transport operator by $L_{\boldsymbol{u}}(\rho) = 0$, meanwhile with the material derivative it is expressed as $\frac{D\rho}{Dt} = -\rho \frac{\partial u^\alpha}{\partial x^\alpha}$.
Now let's go back to Eq. (2.5), and perform the addition and subtraction of term $\frac{\partial (w^\alpha \boldsymbol{U})}{\partial x^\alpha}$ on the left hand side. Then, we write

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial (w^\alpha \boldsymbol{U})}{\partial x^\alpha} + \frac{\partial \boldsymbol{F}_E^\alpha}{\partial x^\alpha} - \frac{\partial (w^\alpha \boldsymbol{U})}{\partial x^\alpha} = \frac{\partial \boldsymbol{D}^\alpha}{\partial x^\alpha} + \boldsymbol{S} \tag{2.15}$$

Using the transport operator definition given in Eq. (2.9), Eq. (2.15) becomes

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \frac{\partial (\boldsymbol{F}_E^\alpha - w^\alpha \boldsymbol{U})}{\partial x^\alpha} = \frac{\partial \boldsymbol{D}^\alpha}{\partial x^\alpha} + \boldsymbol{S} \tag{2.16}$$

Comparison of Eq. (2.5) and Eq. (2.16) summarizes the transformations needed to jump from an Eulerian to an ALE framework. Note that the right hand side of the equations does not suffer any change in the process. In the left hand side of the equations changes occur in pairs. The transient term $\frac{\partial \boldsymbol{U}}{\partial t}$ is replaced by the transport operator $L_{\boldsymbol{w}}(\boldsymbol{U})$ and the Eulerian flux $\boldsymbol{F}_E^\alpha$ is replaced by an ALE flux $\boldsymbol{F}_E^\alpha - w^\alpha \boldsymbol{U}$, which given its dependence with the transport velocity $\boldsymbol{w}$, is denoted as $\boldsymbol{F}_{\boldsymbol{w}}^\alpha$ and reads as

$$\underbrace{\begin{pmatrix} \rho u^\alpha \\ \rho u^\beta u^\alpha + p\delta^{\beta\alpha} \\ (\rho E + p)\, u^\alpha \end{pmatrix}}_{\boldsymbol{F}_E^\alpha} - \underbrace{\begin{pmatrix} \rho w^\alpha \\ \rho u^\beta w^\alpha \\ \rho E w^\alpha \end{pmatrix}}_{w^\alpha \boldsymbol{U}} = \underbrace{\begin{pmatrix} \rho\,(u^\alpha - w^\alpha) \\ \rho u^\beta\,(u^\alpha - w^\alpha) + p\delta^{\beta\alpha} \\ \rho E\,(u^\alpha - w^\alpha) + p u^\alpha \end{pmatrix}}_{\boldsymbol{F}_{\boldsymbol{w}}^\alpha} \tag{2.17}$$

By using the ALE flux $\boldsymbol{F}_{\boldsymbol{w}}^\alpha$ linked with the transport velocity $\boldsymbol{w}$, Eq. (2.16) is rewritten as

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \frac{\partial \boldsymbol{F}^{\alpha}_{\boldsymbol{w}}}{\partial x^{\alpha}} = \frac{\partial \boldsymbol{D}^{\alpha}}{\partial x^{\alpha}} + \boldsymbol{S} \tag{2.18}$$

with the vector of conservative variables $\boldsymbol{U}$, the α-component of the ALE flux $\boldsymbol{F}^{\alpha}_{\boldsymbol{w}}$, the α-component of the diffusive flux $\boldsymbol{D}^{\alpha}$ and the vector of source terms $\boldsymbol{S}$ given by

$$\underbrace{\begin{pmatrix} \rho \\ \rho u^{\beta} \\ \rho E \end{pmatrix}}_{\boldsymbol{U}}, \underbrace{\begin{pmatrix} \rho\left(u^{\alpha} - w^{\alpha}\right) \\ \rho u^{\beta}\left(u^{\alpha} - w^{\alpha}\right) + p\delta^{\beta\alpha} \\ \rho E\left(u^{\alpha} - w^{\alpha}\right) + p u^{\alpha} \end{pmatrix}}_{\boldsymbol{F}^{\alpha}_{\boldsymbol{w}}}, \underbrace{\begin{pmatrix} 0 \\ \tau^{\alpha\beta} \\ k\frac{\partial T}{\partial x^{\alpha}} + \tau^{\alpha\beta}u^{\beta} \end{pmatrix}}_{\boldsymbol{D}^{\alpha}}, \underbrace{\begin{pmatrix} 0 \\ \rho f^{\beta} \\ \rho f^{\beta}u^{\beta} + \dot{q}_h \end{pmatrix}}_{\boldsymbol{S}} \tag{2.19}$$

Equations (2.18) and (2.19) express in vector form the conservative form of the Navier-Stokes equations in ALE framework. With the proper index set for the Greek superscripts, expressions accommodate to 1D, 2D and 3D domains. When the diffusive flux components $\boldsymbol{D}^{\alpha}$ are equal to zero, Euler equations are recovered. Other choices in flux decomposition could be done. Both Eulerian and ALE fluxes account for advective and pressure contributions. Note that when the ALE flux adopts the Lagrangian case ($\boldsymbol{w} = \boldsymbol{u}$) the advective terms are canceled but the pressure contribution remains.

2.3. Principles of SPH method

SPH is a pure meshless method to solve PDE equations by performing a discretization of a continuous domain into a cloud of points.

Given a fluid problem in a continuous domain $\Omega(t)$ expressed in ALE form by

$$\begin{cases} L_{\boldsymbol{w}}(\boldsymbol{U}) + \frac{\partial(\boldsymbol{F}^{\alpha}_{E} - w^{\alpha}\boldsymbol{U})}{\partial x^{\alpha}} = \frac{\partial \boldsymbol{D}^{\alpha}}{\partial x^{\alpha}} + \boldsymbol{S}, & \forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \boldsymbol{U} \in \mathbb{R}^p, t \in \mathbb{R}^+ \\ I.C. : \boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}_0(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \boldsymbol{U}_0 \in \mathbb{R}^p, t = t_0 \\ B.C. : & \forall \boldsymbol{x} \in \partial\Omega \subset \mathbb{R}^d, t > t_0 \end{cases} \tag{2.20}$$

SPH methods look for a discrete solution $\boldsymbol{U}(\boldsymbol{x}_i(t), t)$ on a set of computational points $(P_1, .., P_i, .., P_N)$ moving along the paths following a velocity field $\boldsymbol{w}_i(\boldsymbol{x}_i(t), t)$. Computational points are identified with an integer index i ranging from 1 to N. For a general computational point P_i its initial position is denoted by $\boldsymbol{\chi}_i$ and its current position at time t is given by $\boldsymbol{x}_i(t) = \boldsymbol{x}(\boldsymbol{\chi}_i, t) = x^{\alpha}_i(t) = (x^1_i(t), ..., x^d_i(t))^T$. Time is also discretized in a finite set of time instants $t_0, t_1, t_2, ...t_{final}$.

Figure 2.4 depicts for a two dimensional case the discretization of the continuous domain $\Omega(t)$ and the path of a computational point P_i. Three instants of time t_0, t_1 and t_2 are plotted with $t = t_0$ denoting the time at which initial conditions are prescribed. The movement of points is given by the mapping between computational points with $\boldsymbol{\chi}_i$ coordinates and the spatial coordinates $\boldsymbol{x}_i$. The transport velocity $\boldsymbol{w}_i$ is tangent to the trajectory described by the computational point P_i.

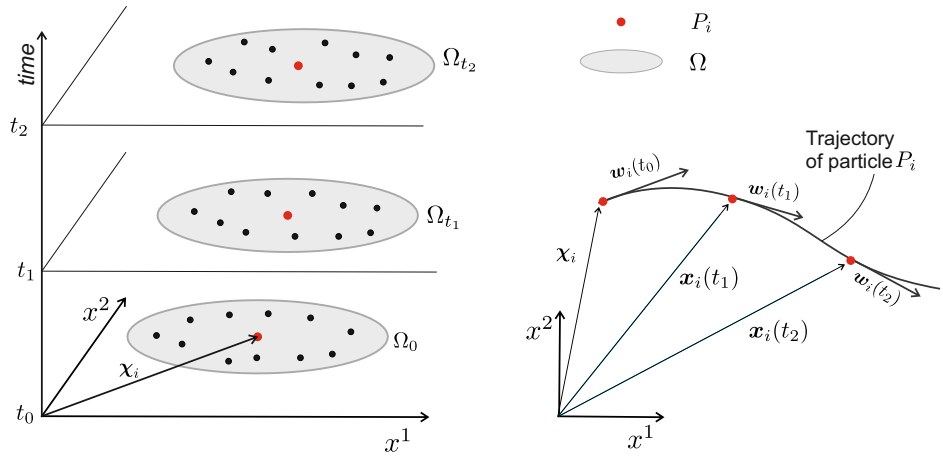


Figure 2.4. On the left: Spatial domain $\Omega(t)$ and set of computational points at three instants of time. On the right: Path of computational point P_i whose initial position is given by χ_i and moves with transport velocity $w_i(t)$.

Computational points are also designed as particles or nodes in literature. In Eulerian description with computational points fixed in space, the term node is usual. In traditional Lagrangian SPH methods, the term particle is preferred. In SPH-ALE methods computational points can be fixed, moving with the fluid velocity or with another desired velocity. Hereafter in this work the term particle will be used in a broad sense and the type of ALE particle is given with the prescribed transport velocity $\boldsymbol{w}$.

A SPH solution of the continuous problem should provide the values of the variables $\boldsymbol{U}_i(t) = \boldsymbol{U}(\boldsymbol{x}_i(t), t)$ on a set of particles $(P_1, ..P_N)$ at a given set of times $t_0, t_1, ...t_{final}$. The SPH procedure could be decomposed into a first stage of initialization and a second stage of evolving the solution in time.

In the initialization stage the discrete set of particles is initialized with the information contained in the initial condition of the continuous solution $\boldsymbol{U}_0(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x}, t = t_0)$. Although it is not strictly necessary, a mesh or other technique such as Voronoi tessellation could be used to provide initial particle weights. This initial weight measurement is associated with the quadrature used for the data points, and in this work is denoted by $V_i(t = t_0)$. Figure 2.5 considers the flow around a circular cylinder to show the main differences between a mesh-based and a meshless method. The top half depicts three instants of time of the grid used by a Lagrangian mesh-based method. The bottom half represents for the same three instants of time the set of points used by a meshless method. We emphasize one cell T_i and one computational particle P_i with filled quadrilateral and red circles. At the initial time $(t = t_0)$ the domain of fluid is rectangular. For a Cartesian layout of the particles we assign an initial volume weight $V_i(t = t_0) = dx \cdot dy$. For $t = t_1$ the set of particles evolves to a new positions dictated

by the transport velocity $\boldsymbol{w}i$. Traditional SPH Lagrangian methods use constant mass weights meanwhile SPH-ALE methods use volume weights. This is further discussed in the next paragraphs.

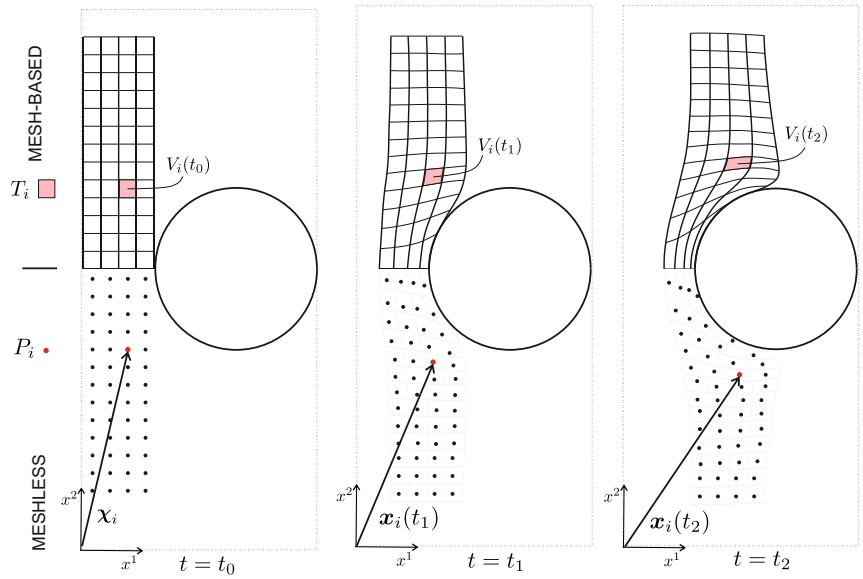


Figure 2.5. Flow around a circular cylinder. Snapshots for three instants of time t_0, t_1, t_2. Discretization corresponding to a mesh-based method in the upper half and to a meshless method on the lower half. The filled color is used to identify the tracking of the same quadrilateral. The red dot identifies the position of particle P_i with initial vector position $\boldsymbol{\chi}_i$.

In the second stage, the values of the variables are advanced in time by solving the discretized governing equations on N computational particles. In this step the set of particle with state $\boldsymbol{U}_i(t = t), V_i(t = t)$ evolve to the next time with state $\boldsymbol{U}_i(t = t + \Delta t), V_i(t = t + \Delta t)$. In order to evolve the state of variables approximating expressions for spatial operators are needed. SPH method requires an approximation technique to obtain discrete versions of spatial operators (gradient, divergence), that must be built from the information available in the particles.

Before explaining the kernel approximation used in SPH, is convenient to introduce the quadrature formula used to estimate the integral of a function $\phi(\boldsymbol{x}, t)$ over a continuous domain $\Omega(t)$. The aim is to obtain an approximation to $\int_{\Omega(t)} \phi(\boldsymbol{x}, t)dx$ with the information available in the set of particles $(P_1, ..P_N)$. Particles are moving following the paths of the transport velocity field $\boldsymbol{w}$. Even though in meshless methods we do not need a mesh, it is useful to invoke the mapping of the mesh to understand the main assumptions used by the quadrature. Figure 2.6 shows a view of three successive positions occupied by particle P_i that at initial time has vector position χ_i and its path

can be expressed by $\boldsymbol{x}_i(t) = \boldsymbol{x}(\chi_i, t)$. In initial time $t = t_0$ the centered square represents the cell associated with particle i that is used to assign its initial volume weight $V_i(t = t_0)$. The deforming shape that result from the mapping of the initial square following the transport velocity field is drawn with dashed lines in times $t = t_1$ and $t = t_2$. For a regular transport velocity field $\boldsymbol{w}$ the volume of particle i for a certain time t is expressed in terms of the Jacobian of the mapping as $V_i(t = t) = J(\chi_i, t = t)V_i(t = t_0)$.

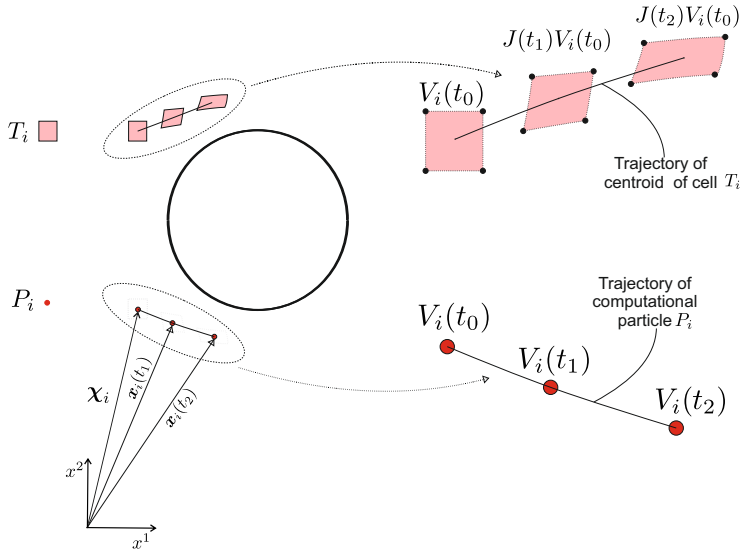


Figure 2.6. Evolution of the weight of a particle P_i along its computational path. On the left: Path of a computational particle P_i moving over a cylinder wall showing its position at three successive times t_0, t_1, t_2 . On the right: Enlarged view showing the evolution of the weights of the particle P_i denoted by $V_i(t_0), V_i(t_1), V_i(t_2)$. The evolution of the weights of particle is compared with the evolution of the area of a quadrilateral that deforms with the same transport velocity field $\boldsymbol{w}$.

The use of the mapping associated to the transport velocity $\boldsymbol{w}$ allows us to refer the integral in the current configuration $\Omega(t)$ to the configuration at initial time $\Omega(t = t_0)$. By assuming a uniform field value $\phi_j(t)$ in the volume $V_j(t)$ associated to each particle j, the following quadrature formula results

$$\int_{\Omega(t)} \phi(\boldsymbol{x}, t)dx = \int_{\Omega(t_0)} \phi(\boldsymbol{x}(\chi, t))J(\chi, t)d\chi \approx \sum_{j \in \Omega(t)} \phi\left(\boldsymbol{x}_j(t)\right) V_j(t) \tag{2.21}$$

where index j runs over j particles $(P_1, ..P_N)$ in $\Omega(t)$. Quadrature formula given by Eq. (2.21) is linked with a particle approximation operator $\Pi(t)$ defined by

$$\Pi(t)(\phi)(\boldsymbol{x}) = \sum_{j \in \Omega(t)} \phi\left(\boldsymbol{x}_j(t)\right) V_j(t)\, \delta\left(\boldsymbol{x} - \boldsymbol{x}_j(t)\right) \tag{2.22}$$

where $\delta\left(\boldsymbol{x} - \boldsymbol{x}_j(t)\right)$ is the Dirac delta function located in $\boldsymbol{x}_j(t)$. Thus, using the Dirac property $\int_{-\infty}^{\infty} \delta\left(\boldsymbol{x} - \boldsymbol{x}_j(t)\right) dx = 1$ we recover Eq. 2.21

$$\int_{\Omega(t)} \phi(\boldsymbol{x}, t) dx \approx \int_{\Omega(t)} \Pi(t)(\phi)(\boldsymbol{x}) dx = \sum_{j \in \Omega(t)} \phi\left(\boldsymbol{x}_j(t)\right) V_j(t) \tag{2.23}$$

So far we have made explicit the dependence of variables with time. Since we are using the *method of lines* [Schiesser, 1991; Schiesser & Griffiths, 2009] to solve the system of PDE, the numerical approach can be decomposed into a first step conducting the spatial discretization and a second step addressing the temporal discretization. Based on this two-step methodology the *method of lines* is also called the *semi-discrete* approach [Hirsch, 2007]. Once the spatial discretization is performed a system of ODEs results that can be integrated with a general time-integration method like explicit and implicit Euler methods or Runge-Kutta methods. Based on the steps involved in the method of lines we omit the time dependence during the spatial discretization process. Eqs. (2.21) and (2.22) are rewritten as

$$\int_{\Omega} \phi(\boldsymbol{x}) dx \approx \sum_{j \in \Omega} \phi\left(\boldsymbol{x}_j\right) V_j \tag{2.24}$$

$$\Pi(\phi)(\boldsymbol{x}) = \sum_{j \in \Omega} \phi\left(\boldsymbol{x}_j\right) V_j\, \delta\left(\boldsymbol{x} - \boldsymbol{x}_j\right) \tag{2.25}$$

The discrete particle approximation given by Eq. (2.22) or Eq. (2.25) gives some guidelines to provide spatial operators for SPH methods, but the Dirac delta is a degenerate function that lacks desirable derivability properties. For that reason, a regularization process is performed by replacing the discontinuous Dirac function by other family of functions with good differentiability properties.

2.3.1. Kernel approximation of a function

Let $(P_1, ., P_i, ..P_N)$ be a set of particles used to discretize a continuous spatial domain Ω. Each particle P_i with barycenter position $\boldsymbol{x}_i$ is linked with a volume weight V_i. Time dependence is omitted since the approximations are spatial operators with fixed time. In Figure 2.7 particles positions for a two-dimensional problem are represented by full filled circles, meanwhile two locations in the continuum domain (denoted as $\boldsymbol{x}$ and $\boldsymbol{x}'$) are represented with hollow circles. We put the focus on two particles with labels i and j and plot concentric circumferences that represent curves of constant value for their kernels.

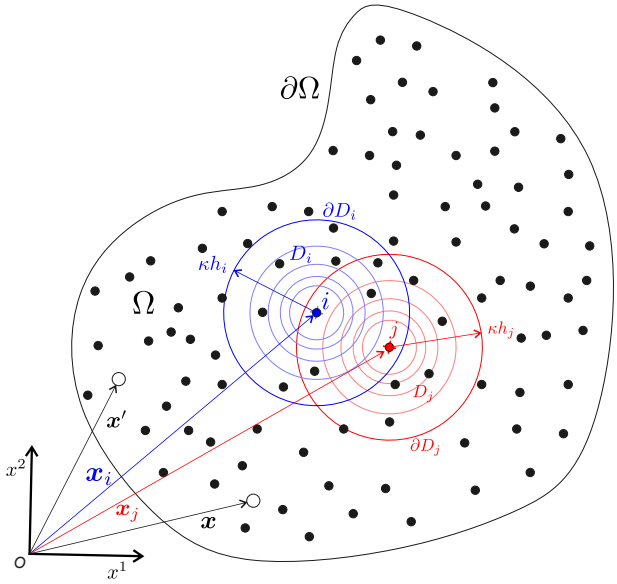


Figure 2.7. Computational domain and kernels for two interacting particles P_i and P_j. Particles positions are represented by full filled circles meanwhile x and x' (hollow circles) are two locations in the continuum domain.

Let $\phi(\boldsymbol{x})$ be a scalar function defined $\forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^d$. The function $\phi(\boldsymbol{x})$ is defined in the whole domain, however in the discrete model only the values in particle set $(P_1, ., P_i, ..P_N)$ are known and denoted as $\phi_i = \phi(\boldsymbol{x}_i)$. Here, approximation of a function ϕ in a point $\boldsymbol{x}$ means an estimation of $\phi(\boldsymbol{x})$ using the information available in a set of particles $\phi_i \equiv \phi(\boldsymbol{x}_i)$.

Convolution formula allows us to express the following identity using the Dirac delta function

$$\phi(\boldsymbol{x}) = \int_\Omega \phi(\boldsymbol{x}') \, \delta(\boldsymbol{x} - \boldsymbol{x}') \, dx'^d \tag{2.26}$$

Despite being an exact expression, the Dirac function does not have convenient properties for derivation. When the delta function $\delta(\boldsymbol{x} - \boldsymbol{x}')$ is replaced by a kernel function $W(\boldsymbol{x} - \boldsymbol{x}', h)$ an integral kernel approximation $\langle \phi(\boldsymbol{x}) \rangle_h^I$ to $\phi(\boldsymbol{x})$ is obtained

$$\langle \phi(\boldsymbol{x}) \rangle_h^I \equiv \int_\Omega \phi(\boldsymbol{x}') \, W(\boldsymbol{x} - \boldsymbol{x}', h) \, dx'^d \tag{2.27}$$

Note that the kernel function comes with a new argument h designated as the smoothing length. In the more general case, the smoothing length could be expressed as function of $h = h(\boldsymbol{x}, \boldsymbol{x}')$, but we first start considering that h is constant in Ω. Approximations obtained with the integral kernel approximation are enclosed in angled brackets, with a subscript h referring to the smoothing length and a superscript I referring to the integral.

However, a second type of approximation must be introduced to evaluate the integral using a quadrature rule. Using the quadrature rule giving in Eq. (2.24) the integral kernel expression Eq. (2.27) becomes the summation kernel approximation

$$\langle \phi(\boldsymbol{x}) \rangle_h = \sum_{j \in \Omega} \phi(\boldsymbol{x}_j) W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j \tag{2.28}$$

Comparison with the particle operator Π given in Eq. (2.25) shows that Eq. (2.28) defines a regularized particle operator where the Dirac function has been replaced by a kernel function. We define the discrete particle approximation of a function ϕ by $\Pi^h(\phi)(\boldsymbol{x})$

$$\Pi^h(\phi)(\boldsymbol{x}) \equiv \sum_{j \in \Omega} \phi(\boldsymbol{x}_j) W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j \tag{2.29}$$

Figure 2.8 plots a kernel function centered in particle j and evaluated at point $\boldsymbol{x}$. We denote as D_j the support domain of particle j and with ∂D_j its boundary. The radius for the support domain D_j is given in terms of the product κh_j, with κ a parameter constant and h_j the smoothing length of particle j.

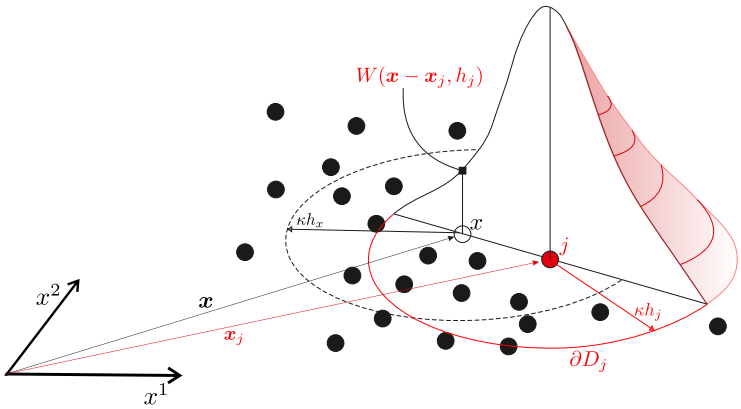


Figure 2.8. Kernel function centered in particle P_j with smoothing lenght h_j in a 2D spatial domain. The support domain of P_j is a circle D_j and its boundary ∂D_j is a circumference of radius κh_j. The evaluation of the kernel centered in P_j in a general position $\boldsymbol{x}$ is expressed by $W(\boldsymbol{x} - \boldsymbol{x}_j, h_j)$.

The smoothing length is a characteristic distance that can be given in terms of the average distance between particles or the weight V_j. There are different types of kernel functions in literature [Dehnen & Aly, 2012]. Kernel functions must satisfy several properties to give satisfactory results in SPH methods: Delta Function limit when $h \to 0$, continuity and derivability, normalization condition, symmetry, positiveness, compact support and monotonous decrease [Price, 2004; Monaghan, 2005]. Liu et al. [Liu et al., 2003a] summarized the most general requirements imposed to kernel functions. Expression $W(\boldsymbol{x} - \boldsymbol{x}_j, h_j)$ should be read as the value of the kernel function

centered in particle j with smoothing length h evaluated at position $\boldsymbol{x}$. Kernel choice for a meshless method has consequences in the stability, convergence, accuracy and computational cost of the numerical scheme. A kernel function is said to be radial when its value depends on the distance between to points. Among all the available radial kernel functions, we choose the cubic spline proposed by Monaghan and Lattanzio [Monaghan & Lattanzio, 1985] given by

$$W_{\boldsymbol{x}'}(\boldsymbol{x}, h) = W\left(\boldsymbol{x} - \boldsymbol{x}', h\right) = W\left(q\right) = \frac{\alpha_d}{h^d} f(q) = \frac{\alpha_d}{h^d} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & q \leqslant 1 \\ \frac{1}{4}(2-q)^3, & 1 < q \leqslant 2 \\ 0, & q > 2 \end{cases} \tag{2.30}$$

where $q = \frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}{h}$ is the normalized distance and α_d is a constant that depends on the dimension d used for assuring the normalization condition of the kernel expressed by $\int_\Omega W\left(\boldsymbol{x} - \boldsymbol{x}', h\right) dx'^d = 1$. The normalization condition for the kernel is equivalent to the enforcement of the partition of unity for the integral kernel approximation. For 1D problems $\alpha_1 = \frac{2}{3}$, for 2D problems $\alpha_2 = \frac{10}{7\pi}$ and for 3D $\alpha_3 = \frac{1}{\pi}$. Kernel given by Eq. (2.30) satisfies the set of desirable properties for a smoothing kernel function and it is plotted in Figure 2.9 for the 1D case.

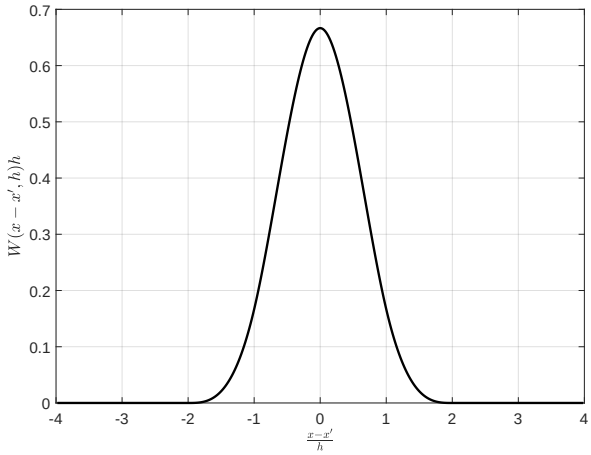


Figure 2.9. Kernel function of the 1D cubic spline, given by Eq. (2.30)**.**

Since the kernel function has a compact support, it is possible to reduce the integration over the whole domain Ω to the compact support of a particle ($D(x)$).

$$\langle \phi(\boldsymbol{x}) \rangle_h^I = \int_\Omega \phi\left(\boldsymbol{x}'\right) W\left(\boldsymbol{x} - \boldsymbol{x}', h\right) dx'^d = \int_{D(\boldsymbol{x})} \phi\left(\boldsymbol{x}'\right) W\left(\boldsymbol{x} - \boldsymbol{x}', h\right) dx'^d \tag{2.31}$$

Similarly, for the discrete kernel approximation, instead of summing over all particles $(P_1, ., P_i, ..P_N)$ in the domain Ω, is enough to sum over the neighbor particles inside the compact support $D(x)$.

$$\langle \phi(\boldsymbol{x}) \rangle_h = \sum_{j \in \Omega} \phi(\boldsymbol{x}_j) W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j = \sum_{j \in D(\boldsymbol{x})} \phi(\boldsymbol{x}_j) W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j \tag{2.32}$$

2.3.2. Kernel approximation of the gradient of a function

In order to obtain the kernel approximation of the gradient of a function ϕ, we replace ϕ by $\nabla\phi$ in Eq. (2.31) to obtain

$$\langle \nabla\phi(\boldsymbol{x}) \rangle_h^I = \int_{D(\boldsymbol{x})} \nabla_{\boldsymbol{x}'} \phi(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \, dx'^d \tag{2.33}$$

where we use the compact support property of the kernel to define the integration region as $D(\boldsymbol{x})$ instead of Ω. We include subscript $\boldsymbol{x}'$ in the nabla operator to emphasize that the gradient in the convolution is taken with respect to the primed variable. Using integration by parts we get

$$\langle \nabla\phi(\boldsymbol{x}) \rangle_h^I = \int_{D(\boldsymbol{x})} \nabla_{\boldsymbol{x}'} \big(\phi(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \big) dx'^d - \int_{D(\boldsymbol{x})} \phi(\boldsymbol{x}') \nabla_{\boldsymbol{x}'} W(\boldsymbol{x} - \boldsymbol{x}', h) \, dx'^d \tag{2.34}$$

that after applying of Gauss theorem in the first integral, using the gradient relation for symmetric kernels $\nabla_{\boldsymbol{x}'} W(\boldsymbol{x} - \boldsymbol{x}', h) = -\nabla_{\boldsymbol{x}} W(\boldsymbol{x} - \boldsymbol{x}', h)$ and reordering yields

$$\langle \nabla\phi(\boldsymbol{x}) \rangle_h^I = \int_{D(\boldsymbol{x})} \phi(\boldsymbol{x}') \nabla_{\boldsymbol{x}} W(\boldsymbol{x} - \boldsymbol{x}', h) \, dx'^d + \int_{\partial D(x)} \phi(\boldsymbol{x}') W(\boldsymbol{x} - \boldsymbol{x}', h) \boldsymbol{n}(\boldsymbol{x}') \, dx'^{d-1} \tag{2.35}$$

where $\boldsymbol{n}$ stands for the unit vector pointing in the normal outward to the boundary support ∂D. Eq. (2.35) provides the integral kernel approximation for the gradient of a scalar function ϕ. When the support of the kernel $D(\boldsymbol{x})$ does not intersect the boundary of the domain $\partial\Omega$, the boundary integral term in Eq. (2.35) is equal to zero, due to the compact support of the kernel.

When the integral is approximated with the quadrature rule Eq. (2.24), the summation kernel gradient approximation reads:

$$\langle \nabla\phi(\boldsymbol{x}) \rangle_h = \sum_{j \in D(\boldsymbol{x})} \phi_j \nabla_{\boldsymbol{x}} W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j + \sum_{j \in \partial D(\boldsymbol{x})} \phi_j W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j^{\partial} \boldsymbol{n}_j \tag{2.36}$$

where V_j^{∂} stands for the area/length of the boundary element dx'^{d-1}

In terms of the regularized particle operator Π^h, the approximation for the gradient of a function ϕ at a point $\boldsymbol{x}$ is written as $\langle \nabla\phi(\boldsymbol{x}) \rangle_h = \Pi^h(\nabla\phi)(\boldsymbol{x})$

When the support of the kernel $D(x)$ is far from the boundary $\partial\Omega$, the sum over $\partial D(x)$ is zero and we can change the order of the approximation and gradient operator

$$\langle\nabla\phi(\boldsymbol{x})\rangle_h = \Pi^h(\nabla\phi)(\boldsymbol{x}) = \sum_{j\in D(\boldsymbol{x})} \phi_j \nabla_{\boldsymbol{x}} W\left(\boldsymbol{x}-\boldsymbol{x}_j, h\right) V_j = \nabla\left(\Pi^h(\phi)(\boldsymbol{x})\right) = \nabla\langle\phi(\boldsymbol{x})\rangle_h \tag{2.37}$$

The nice feature about kernel approximation of derivatives is that it only requires the determination of the term $\nabla_{\boldsymbol{x}} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right)$, that can be obtained analytically. For the case of radial kernels and assuming constant smoothing length h, we obtain

$$\nabla_{\boldsymbol{x}} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) = \frac{\partial W\left(\boldsymbol{x}-\boldsymbol{x}', h\right)}{\partial \boldsymbol{x}} = \frac{\partial W\left(\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|, h\right)}{\partial \boldsymbol{x}} = \frac{\partial W\left(q\right)}{\partial \boldsymbol{x}} \tag{2.38}$$

For radial kernels expressed in the form $W(q) = \frac{\alpha_d}{h^d} f(q)$, with $q = \frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}{h}$, application of the chain rule allows us to obtain a more meaningful expression for the kernel gradient

$$\frac{\partial W\left(q\right)}{\partial \boldsymbol{x}} = \frac{\alpha_d}{h^d}\frac{\partial f(q)}{\partial \boldsymbol{x}} = \frac{\alpha_d}{h^d}\frac{df(q)}{dq}\frac{\partial q}{\partial \boldsymbol{x}} = \frac{\alpha_d}{h^d}\frac{df(q)}{dq}\frac{1}{h}\frac{\partial\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}{\partial \boldsymbol{x}} = \frac{\alpha_d}{h^{d+1}}\frac{df(q)}{dq}\frac{\left(\boldsymbol{x}-\boldsymbol{x}'\right)}{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|} \tag{2.39}$$

where the fraction $\frac{\left(\boldsymbol{x}-\boldsymbol{x}'\right)}{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}$ is the unit vector pointing from position $\boldsymbol{x}'$ to $\boldsymbol{x}$. As function $f(q)$ is decreasing, its derivative $\frac{df(q)}{dq}$ is negative. These properties are important to interpret the kernel gradient as a vector with modulus and direction. For the cubic spline kernel, the following expression for the kernel gradient is obtained

$$\nabla_{\boldsymbol{x}} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) = \frac{\alpha_d}{h^{d+1}}\frac{df}{dq}\frac{\boldsymbol{x}-\boldsymbol{x}'}{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|} = \frac{\alpha_d}{h^{d+1}}\left\{\begin{array}{cc} -3q+\frac{9}{4}q^2 & q\leqslant 1 \\ -\frac{3}{4}(2-q)^2 & 1<q\leqslant 2 \\ 0 & q>2 \end{array}\right\}\frac{\boldsymbol{x}-\boldsymbol{x}'}{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|} \tag{2.40}$$

that is plotted in Figure 2.10 for the 1D case.

2.3.3. Accuracy and consistency of kernel approximation

Integral kernel approximation

Performing a Taylor series expansion of $\phi(\boldsymbol{x}')$ around $\boldsymbol{x}$ and inserting it in Eq. (2.27) gives

$$\begin{aligned}\langle\phi(\boldsymbol{x})\rangle_h^I = \int_{D(x)} \Bigg[&\phi(\boldsymbol{x}) + (x'^{\alpha}-x^{\alpha})\frac{\partial\phi}{\partial x^{\alpha}} \\ &+ \frac{1}{2}(x'^{\alpha}-x^{\alpha})(x'^{\beta}-x^{\beta})\frac{\partial^2\phi}{\partial x^{\alpha}\partial x^{\beta}} + \mathcal{O}\left(\left(\boldsymbol{x}-\boldsymbol{x}'\right)^3\right)\Bigg] W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d\end{aligned} \tag{2.41}$$

We recall the following properties of the kernel:

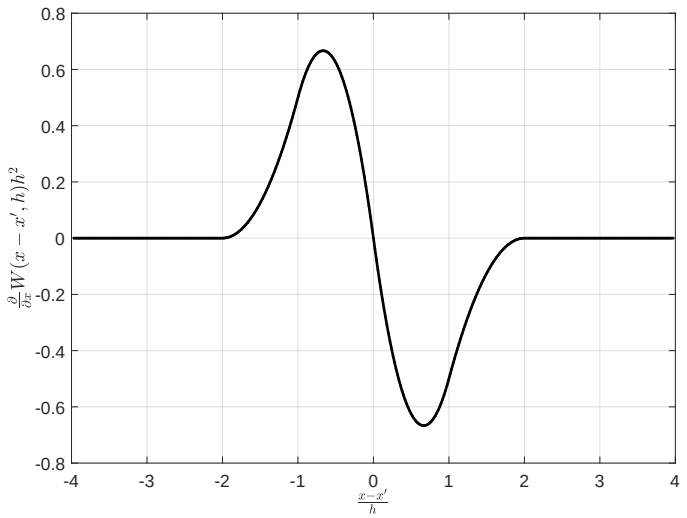


Figure 2.10. Kernel gradient of the 1D cubic spline, given by Eq. (2.40)**.**

- normalized kernel: $\int_{D(\boldsymbol{x})} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d = 1$
- symmetric kernel: $W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) = W\left(\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|, h\right) = W\left(\boldsymbol{x}'-\boldsymbol{x}, h\right)$ and therefore $\int_{D(\boldsymbol{x})} \left(\boldsymbol{x}'-\boldsymbol{x}\right) W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d = \mathbf{0}$

Then, Eq. (2.41) reduces to

$$\langle\phi(\boldsymbol{x})\rangle_h^I = \phi(\boldsymbol{x}) + \frac{1}{2}\frac{\partial^2\phi}{\partial x^\alpha \partial x^\beta}\int_{D(\boldsymbol{x})}(x'^\alpha - x^\alpha)(x'^\beta - x^\beta)W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d + \mathcal{O}\left[\left(\boldsymbol{x}-\boldsymbol{x}'\right)^4\right] \tag{2.42}$$

The integrand appearing in the second term of the right hand side can be bounded by the squared radius of the kernel support $(\kappa h)^2$ and therefore it is shown that the integral kernel interpolation is second order accurate $[\mathcal{O}(h^2)]$

$$\langle\phi(\boldsymbol{x})\rangle_h^I = \phi(\boldsymbol{x}) + \mathcal{O}\left(h^2\right) \tag{2.43}$$

The order of consistency of an approximation is defined as the highest order of a polynomial that can be reproduced exactly. Eq. (2.43) shows that the kernel approximation has first-order consistency, meaning that integral kernel approximation can reproduce exactly constant and linear polynomials.

The errors resulting from the integral kernel approximation for the gradient may be estimated similarly. Assuming that the point $\boldsymbol{x}$ is far away from the boundary $\partial\Omega$ inserting the Taylor series expansion of $\phi(\boldsymbol{x}')$ around $\boldsymbol{x}$ in Eq. (2.35) gives

$$\langle \nabla\phi(\boldsymbol{x})\rangle_h^I = \int_{D(\boldsymbol{x})} \Big[\phi(\boldsymbol{x}) + (x'^\beta - x^\beta)\frac{\partial\phi}{\partial x^\beta} + \frac{1}{2}(x'^\beta - x^\beta)(x'^\gamma - x^\gamma)\frac{\partial^2\phi}{\partial x^\beta \partial x^\gamma} + \mathcal{O}\left((\boldsymbol{x}-\boldsymbol{x}')^3\right)\Big] \nabla_{\boldsymbol{x}} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d \quad (2.44)$$

For symmetric kernels the integral of the first and third terms of the expansion are zero, and noting that the integral of the $\mathcal{O}\left((\boldsymbol{x}-\boldsymbol{x}')^3\right)$ scales with $\mathcal{O}(h^2)$ ($\int \nabla W \mathrm{d}x'^d$ scales with $1/h$) we obtain for each α component

$$\left\langle \frac{\partial}{\partial x^\alpha}\phi(\boldsymbol{x})\right\rangle_h^I = \frac{\partial\phi}{\partial x^\beta}\int_{D(\boldsymbol{x})}(x'^\beta - x^\beta)\frac{\partial}{\partial x^\alpha} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d + \mathcal{O}(h^2) \quad (2.45)$$

Using integration by parts, it can be proved that far from the boundary the integral equals the identity matrix

$$\int_{D(\boldsymbol{x})}(x'^\beta - x^\beta)\frac{\partial}{\partial x^\alpha} W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d = \int_{\partial D(\boldsymbol{x})}(x'^\beta - x^\beta) W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) \boldsymbol{n}' dx'^{d-1} - \int_{D(\boldsymbol{x})}\frac{\partial}{\partial x^\alpha}(x'^\beta - x^\beta) W\left(\boldsymbol{x}-\boldsymbol{x}', h\right) dx'^d = \delta^{\alpha\beta} \quad (2.46)$$

and the integral kernel approximation for the gradient provides approximations with second order accuracy

$$\langle \nabla\phi(\boldsymbol{x})\rangle_h^I = \nabla\phi(\boldsymbol{x}) + \mathcal{O}(h^2) \quad (2.47)$$

Thus, it can be concluded that far from the boundary, a symmetric kernel normalized with compact support can provide approximations for regular functions and its gradients with second order accuracy.

Summation kernel approximation

Satisfying some kernel consistency conditions does not necessarily mean that the discretized form also satisfies the discretized consistency condition. This discrepancy between the particle approximation and the kernel approximation is termed as particle inconsistency. The discrete counterparts of the constant and linear consistency conditions are

$$\sum_{j\in D(\boldsymbol{x})} W(\boldsymbol{x}-\boldsymbol{x}_j, h)V_j = 1 \quad (2.48)$$

$$\sum_{j\in D(\boldsymbol{x})} (\boldsymbol{x}-\boldsymbol{x}_j) W(\boldsymbol{x}-\boldsymbol{x}_j, h)V_j = 0 \quad (2.49)$$

These discretized consistency conditions are not always satisfied, even for uniform particle distributions. For that reason, several alternatives expressions for the discrete gradient approximations were proposed in the literature [Liu & Liu, 2006] [Magoules, 2011]. For example, restoring zero-order consistency for the approximation of the gradient can be enforced by defining the discrete gradient operator $D_h(\phi)(\boldsymbol{x})$ by

$$D_h(\phi)(\boldsymbol{x}) \equiv \Pi^h(\nabla\phi)(\boldsymbol{x}) - \phi(\boldsymbol{x})\Pi^h(\nabla 1)(\boldsymbol{x}) = \sum_{j\in D(\boldsymbol{x})} (\phi_j - \phi(\boldsymbol{x}))\, \nabla_{\boldsymbol{x}} W\,(\boldsymbol{x}-\boldsymbol{x}_j, h)\, V_j \tag{2.50}$$

With this correction the zero consistency is recovered, despite the discrete condition $\sum_{j\in D(\boldsymbol{x})} \nabla_{\boldsymbol{x}} W\,(\boldsymbol{x}-\boldsymbol{x}_j, h)\, V_j = 0$ is not fulfilled exactly.

Recovering first order consistency for a kernel gradient operator is also possible although it requires a more elaborated strategy [Liu & Liu, 2006].

2.3.4. Meshless and mesh-based approximations

The need for an approximation technique for meshless methods like SPH is not different to the need for shape functions for Finite Element Method (FEM). The difference only resides in the geometric information contained in a cloud of points and the information contained in a mesh of elements. A brief comparison of kernel approximation and meshless interpolation is useful to clarify some problematic issues with notation in SPH.

Figure 2.11 shows a comparison between kernel function W used by SPH (left) and the shape function N used by the FEM (right).

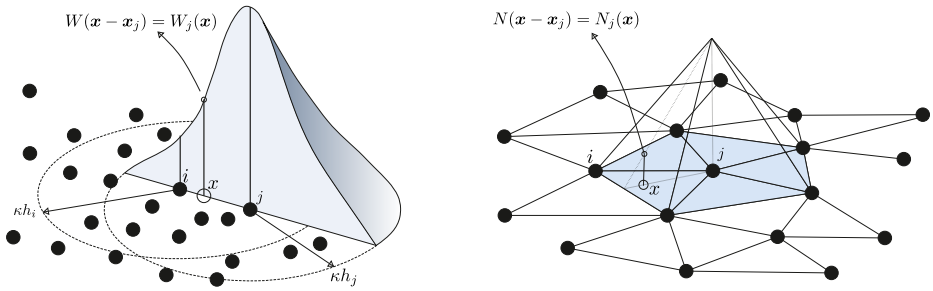


Figure 2.11. SPH kernel function $W(\boldsymbol{x}-\boldsymbol{x}_j)$ and FEM shape function $N(\boldsymbol{x}-\boldsymbol{x}_j)$. The support of the kernel function of P_j is specified in terms of its smoothing length h_j meanwhile in FEM methods the support of the shape functions is given in terms of the elements that share the node j as a vertex.

We clarify the nomenclature used for both the kernel and shape function. The kernel function centered in $\boldsymbol{x}_i$ and evaluated at point $\boldsymbol{x}$ is represented by $W(\boldsymbol{x}-\boldsymbol{x}_i, h)$ and in compact form with $W_i(\boldsymbol{x}, h)$. The same is done for the basis function and we denote $N(\boldsymbol{x}-\boldsymbol{x}_i)$ the base function centered in node $\boldsymbol{x}_i$ and evaluated in point $\boldsymbol{x}$.

In SPH the kernel approximation of a scalar u in a point $\boldsymbol{x}$ is given by

$$\langle u(\boldsymbol{x})\rangle = \sum_{j=1}^{n} u\left(\boldsymbol{x}_j\right) W\left(\boldsymbol{x}-\boldsymbol{x}_j, h\right) V_j \tag{2.51}$$

In the Finite Element Method a scalar function u is approximated by a given set of functions N_i according to

$$\hat{u}(\boldsymbol{x}) = \sum_{j=1}^{n} N(\boldsymbol{x}-\boldsymbol{x}_j) u\left(\boldsymbol{x}_j\right) \tag{2.52}$$

By comparison of previous equations it can be shown that the weights of nodes in FEM ($N_j(x)$) are expressed as $W(\boldsymbol{x}-\boldsymbol{x}_j, h)V_j$ in SPH nomenclature, that is

$$N(\boldsymbol{x}-\boldsymbol{x}_j) \approx W(\boldsymbol{x}-\boldsymbol{x}_j, h)V_j \tag{2.53}$$

It must be noticed that there are some remarkable differences between FEM shape functions and kernel functions. Kernel functions use a second argument (h) to delimitate the support extension, whereas with shape functions meanwhile the extension is defined implicitly with the element. Shape functions are dimensionless, meanwhile the kernel function has the inverse units of the volume weight (V_j). For shape functions the number of nodes n can be inferred from the element, whereas in SPH it depends on the results of the neighbor searching algorithm.

2.4. SPH-ALE derivation

Here we present the derivation of the SPH-ALE method according to the work conducted by Vila [Vila, 1999]. The method was proposed for a general system of conservation laws with a main focus on the Euler equations. Bounded domains were considered as extensions of the method and therefore boundary conditions were excluded in the derivation process. Since then, the SPH-ALE method was improved and extended by Vila et al. ([Ben Moussa & Vila, 2000], [Mancip, 2001], [Vila, 2005], [Cueille, 2005], [Lanson & Vila, 2008a], [Collé et al., 2019]) and authors of other researching groups ([Marongiu, 2007],[Renaut, 2015],[Avesani et al., 2015], [Nogueira et al., 2016a], [Eirís et al., 2021]). In some of the works using the SPH-ALE some alternative derivations are presented, ranging from elegant ones using the partition of unity concept [Ivanova et al., 2013] to fast ones using the analogy with the Finite Volume Method [Oger et al., 2016].

The set of partial differential equations considered in this section are the Euler equations. By dropping the viscous terms in Eq. (2.18) the conservative ALE form for the Euler equations can be compactly expressed by

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \frac{\partial \boldsymbol{F}^{\alpha}_{\boldsymbol{w}}}{\partial x^{\alpha}} = \boldsymbol{S} \tag{2.54}$$

with the transport operator L_w defined in Eq. (2.9).

The Initial Value Problem (IVP) considered by Vila is expressed by

$$\begin{cases} L_{\boldsymbol{w}}(\boldsymbol{U})(\boldsymbol{x},t) + \frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}(\boldsymbol{x},t) = \boldsymbol{S}(\boldsymbol{x},t), & \forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \boldsymbol{U} \in \mathbb{R}^p, t \in \mathbb{R}^+ \\ \boldsymbol{U}(\boldsymbol{x},0) = \boldsymbol{U}_0(\boldsymbol{x}), & \forall \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \boldsymbol{U}_0 \in \mathbb{R}^p, t = t_0 \end{cases} \tag{2.55}$$

For the same system of equations, it is possible to look for a strong solution or a weak solution.

2.4.1. Strong solution

The procedure to obtain a strong solution for the IVP expressed in Eq. (2.55) can be obtained with the tools presented in previous section. For the sake of clarify, Table (2.1) collects the definition of Dirac particle operator $\Pi(t)$, the regularized or kernel particle operator $\Pi^h(t)$ and the regularized gradient operator $\nabla\Pi^h(t)$.

Operator	Definition
$\Pi(t)(\phi)(\boldsymbol{x})$	$\sum_{j\in\Omega(t)} \phi(\boldsymbol{x}_j(t))\,\delta(\boldsymbol{x}-\boldsymbol{x}_j(t))\,V_j(t)$
$\Pi^h(t)(\phi)(\boldsymbol{x})$	$\sum_{j\in\Omega(t)} \phi(\boldsymbol{x}_j(t))\,W(\boldsymbol{x}-\boldsymbol{x}_j(t),h)\,V_j(t)$
$\nabla\Pi^h(t)(\phi)(\boldsymbol{x})$	$\sum_{j\in\Omega(t)} \phi(\boldsymbol{x}_j(t))\,\nabla_{\boldsymbol{x}}W(\boldsymbol{x}-\boldsymbol{x}_j(t),h)\,V_j(t)$

Table 2.1. Definitions of Particle Operator $\Pi(t)(\cdot)(\boldsymbol{x})$, Regularized Particle Operator $\Pi^h(t)(\cdot)(\boldsymbol{x})$ and Gradient Particle Operator $\nabla\Pi^h(t)(\cdot)(\boldsymbol{x})$ acting on a general scalar variable ϕ using Vila notation [Vila, 1999].

We recall the relation of the particle quadrature with particle operator $\Pi(t)$ given in Eq. (2.23)

$$\int_{\Omega(t)} \phi(\boldsymbol{x},t)dx \approx \int_{\Omega(t)} \Pi(t)(\phi)(\boldsymbol{x})dx = \sum_{j\in\Omega(t)} \phi(\boldsymbol{x}_j(t))\,V_j(t) \tag{2.56}$$

When the particle quadrature is applied component wise for a vector function $\boldsymbol{\Phi}(\boldsymbol{x})$ reads

$$\int_{\Omega(t)} \boldsymbol{\Phi}(\boldsymbol{x},t)dx \approx \int_{\Omega(t)} \Pi(t)(\boldsymbol{\Phi})(\boldsymbol{x})dx = \sum_{j\in\Omega(t)} \boldsymbol{\Phi}(\boldsymbol{x}_j(t))\,V_j(t) \tag{2.57}$$

Application of the particle quadrature given by Eq. (2.57) to a vector function $\boldsymbol{\Phi}(\boldsymbol{x})$ $= L_{\boldsymbol{w}}(\boldsymbol{U})(\boldsymbol{x},t) + \frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}(\boldsymbol{x},t) - \boldsymbol{S}(\boldsymbol{x},t) = \boldsymbol{0}$ leads to

$$\begin{aligned} \boldsymbol{0} \approx \int_{\Omega(t)} \Pi(t)(L_{\boldsymbol{w}}(\boldsymbol{U}))(\boldsymbol{x})dx + \int_{\Omega(t)} \Pi(t)\left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)(\boldsymbol{x})dx - \int_{\Omega(t)} \Pi(t)(\boldsymbol{S})(\boldsymbol{x})dx = \\ \sum_{j\in\Omega(t)} L_{\boldsymbol{w}}(\boldsymbol{U})(\boldsymbol{x}_j(t))V_j(t) + \sum_{j\in\Omega(t)} \frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}(\boldsymbol{x}_j(t))V_j(t) - \sum_{j\in\Omega(t)} \boldsymbol{S}(\boldsymbol{x}_j(t))V_j(t) \end{aligned} \tag{2.58}$$

Since we are looking for a strong solution, we force that every particle i satisfies the relation

$$L_{\boldsymbol{w}}(\boldsymbol{U})(\boldsymbol{x}_i(t))V_i(t) + \frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}(\boldsymbol{x}_i(t))V_i(t) - \boldsymbol{S}(\boldsymbol{x}_i(t))V_i(t) = \boldsymbol{0} \tag{2.59}$$

that, using the compact notation $\phi_i(t) = \phi(\boldsymbol{x}_i(t))$ and dropping the time dependence, is rewritten as

$$\left(L_{\boldsymbol{w}}(\boldsymbol{U})\right)_i V_i + \left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)_i V_i - \boldsymbol{S}_i V_i = \boldsymbol{0} \tag{2.60}$$

The use of the Generalized Reynolds Transport (See Eq. (2.10)) allows us to identify the first term as $\left(L_{\boldsymbol{w}}(\boldsymbol{U})\right)_i V_i = \frac{d}{dt}\left(\boldsymbol{U}_i V_i\right)$ yielding to

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + \left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)_i V_i = \boldsymbol{S}_i V_i \tag{2.61}$$

Once the Dirac particle approximation for the three terms has been used, the next step requires the kernel approximation for approximating the divergence of the ALE flux as

$$\left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)_i \simeq \left\langle\left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)_i\right\rangle^h = \frac{\partial}{\partial x^\alpha}\Pi^h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i = \sum_{j\in\Omega(t)}(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j \frac{\partial}{\partial x^\alpha_i} W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) V_j \tag{2.62}$$

that after substitution in Eq. (2.61), gives the final expression

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i \sum_{j\in\Omega(t)}(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j \frac{\partial}{\partial x^\alpha_i} W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) V_j = \boldsymbol{S}_i V_i \tag{2.63}$$

Using the compact discrete designation for the kernel $\nabla^\alpha_{\boldsymbol{x}_i} W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) = \nabla^\alpha_i W_{ij}$, the following semi-discrete system of equations is obtained for the strong SPH-ALE semi-discrete form

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i \sum_{j\in\Omega(t)}(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j \nabla^\alpha_i W_{ij} V_j = \boldsymbol{S}_i V_i \tag{2.64}$$

However, semi-discrete system given by Eq. (2.64) does not assure the local conservation property. The condition for local conservation demands that the sum of the ALE flux between two neighboring particles satisfies

$$V_i(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i \frac{\partial}{\partial x^\alpha_i} W_{ij} V_j + V_j(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j \frac{\partial}{\partial x^\alpha_j} W_{ji} V_j = 0 \tag{2.65}$$

Using the symmetry of the kernel we have that $\frac{\partial}{\partial x^\alpha_i} W_{ij} = -\frac{\partial}{\partial x^\alpha_j} W_{ji}$ but since usually $(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i \neq (\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j$ local conservation is violated.

It is possible to select a symmetric gradient approximation to circumvent the violation of local conservation. A kernel gradient approximation defined by $\nabla \Pi^h(\phi)(\boldsymbol{x}) + \phi(\boldsymbol{x})\nabla \Pi^h(1)$ would approximate the gradient as

$$\left(\frac{\partial}{\partial x^\alpha}\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right)_i \simeq D^\alpha_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i = \sum_{j\in\Omega(t)} \left((\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i + (\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j\right) \frac{\partial}{\partial x^\alpha_i} W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) V_j \tag{2.66}$$

giving the alternative semi-discrete system

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i \sum_{j\in\Omega(t)} \left((\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i + (\boldsymbol{F}^\alpha_{\boldsymbol{w}})_j\right) \frac{\partial}{\partial x^\alpha_i} W_{ij} V_j = \boldsymbol{S}_i V_i \tag{2.67}$$

We have presented the derivation of the strong form of the IVP given by Eq. (2.55). The procedure does not provide an unique semi-discrete system. For instance, both semi-discrete systems given by Eq. (2.64) and Eq. (2.67) were obtained by adopting different kernel approximations for the gradient.

2.4.2. Weak SPH solution

In order to deal with nonlinear hyperbolic conservation laws and the computation of discontinuous solutions, it is necessary to introduce the notion of a weak solution. Here we seek a discrete formulation of meshless equations by starting from a weak form of conservation equations. First, we tackle the continuous weak form of conservation law given in Eq. (2.54) and then, we develop the discretization over the set of computational points.

Continuous weak form

To obtain the weak form, first suppose that $\boldsymbol{U}(\boldsymbol{x}, t)$ is a smooth solution of Eq. (2.54). By expanding the transport operator, the system of PDEs for the Euler equations in ALE form reads

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial}{\partial x^\alpha}\left(w^\alpha \boldsymbol{U}\right) + \frac{\partial}{\partial x^\alpha}\left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right) - \boldsymbol{S} = \boldsymbol{0} \tag{2.68}$$

We multiply Eq. (2.68) by a set of test functions $\varphi(\boldsymbol{x}, t)$ with compact support, and then, we perform the integration over $\mathbb{R}^d \times \mathbb{R}^+$

$$\int_{\mathbb{R}^d\times\mathbb{R}^+} \left(\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial}{\partial x^\alpha}\left(w^\alpha \boldsymbol{U}\right) + \frac{\partial}{\partial x^\alpha}\left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}\right) - \boldsymbol{S}\right) \varphi(x,t) dx^d dt = \boldsymbol{0} \tag{2.69}$$

We decompose the integral in order to separate the time and spatial partial derivatives. The compact support of the test function allows us to change the order in the integration over space or time, obtaining

$$\int_{\mathbb{R}^+\times\mathbb{R}^d} \frac{\partial \boldsymbol{U}}{\partial t}\varphi dt\, dx^d + \int_{\mathbb{R}^d\times\mathbb{R}^+} \left(\frac{\partial\left(w^\alpha \boldsymbol{U}\right)}{\partial x^\alpha}\varphi + \frac{\partial \boldsymbol{F}^\alpha_{\boldsymbol{w}}}{\partial x^\alpha}\varphi - S\varphi\right) dx^d\, dt = \boldsymbol{0} \tag{2.70}$$

using integration by parts, we develop both integrals to obtain

$$
\begin{aligned}
&-\int_{\mathbb{R}^d}\varphi(x,0)\boldsymbol{U}(x,0)dx^d-\int_{\mathbb{R}^d\times\mathbb{R}^+}\boldsymbol{U}(x,t)\frac{\partial\varphi}{\partial t}(x,t)dx^d dt\\
&\quad+\int_{\mathbb{R}^d\times\mathbb{R}^+}\left(-w^\alpha\boldsymbol{U}\frac{\partial\varphi}{\partial x^\alpha}+\frac{\partial\left(w^\alpha\boldsymbol{U}\varphi\right)}{\partial x^\alpha}-\boldsymbol{F}^\alpha_{\boldsymbol{w}}\frac{\partial\varphi}{\partial x^\alpha}+\frac{\partial\left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}\varphi\right)}{\partial x^\alpha}-S\varphi\right)dx^d\,dt=0
\end{aligned}
\tag{2.71}
$$

that can be rearranged by moving the initial condition term to the right hand side and invoking Gauss theorem to give

$$
\begin{aligned}
&\int_{\mathbb{R}^d\times\mathbb{R}^+}\left(-\boldsymbol{U}\frac{\partial\varphi}{\partial t}-w^\alpha\boldsymbol{U}\frac{\partial\varphi}{\partial x^\alpha}-\boldsymbol{F}^\alpha_{\boldsymbol{w}}\frac{\partial\varphi}{\partial x^\alpha}-S\varphi\right)dx^d\,dt\\
&\qquad+\int_{\partial\mathbb{R}^d\times\mathbb{R}^+}\left(w^\alpha\boldsymbol{U}\varphi+\boldsymbol{F}^\alpha_{\boldsymbol{w}}\varphi\right)n^\alpha dx^{d-1}\,dt=\int_{\mathbb{R}^d}\varphi(\boldsymbol{x},0)\boldsymbol{U}(\boldsymbol{x},0)\,dx^d
\end{aligned}
\tag{2.72}
$$

The boundary term in the frontier of $\mathbb{R}^d$ vanishes since the test function $\varphi(\boldsymbol{x},t)$ are zero outside the domain $\Omega(t)$, and $\Omega(t)$ is a bounded region of space $\mathbb{R}^d$. In the first integral, we note the presence of the fictitious material derivative of the test function $\frac{\partial\varphi}{\partial t}+w^\alpha\frac{\partial\varphi}{\partial x^\alpha}$. This fictitious attribute is used to remark that the advective contribution to the derivative is due to the transport velocity $\boldsymbol{w}$, and not to the true fluid velocity $\boldsymbol{u}$, as it is the usual material derivative in Fluid Mechanics. Introducing the operator $L^*_{\boldsymbol{w}}(\cdot)\equiv\frac{\partial(\cdot)}{\partial t}+w^\alpha\frac{\partial(\cdot)}{\partial x^\alpha}$, we refer to this term as $L^*_{\boldsymbol{w}}(\varphi)$. Taking into account these considerations, the previous equation becomes

$$
\int_{\mathbb{R}^d\times\mathbb{R}^+}\left(\boldsymbol{U}L^*_{\boldsymbol{w}}(\varphi)+\boldsymbol{F}^\alpha_{\boldsymbol{w}}\frac{\partial\varphi}{\partial x^\alpha}+\boldsymbol{S}\varphi\right)dx^d\,dt=-\int_{\mathbb{R}^d}\varphi(\boldsymbol{x},0)\boldsymbol{U}(\boldsymbol{x},0)\,dx^d
\tag{2.73}
$$

We have arrived to an integral equation, which is also verified for a smooth solution of Eq. (2.68) for every test function φ with compact support. Note that there are no derivatives of $\boldsymbol{U}$ in Eq. (2.73). Reciprocally, if we suppose that a smooth function $\boldsymbol{U}$ satisfies Eq. (2.73) for every test function φ Eq. (2.73), integration by parts in the reverse order shows that $\boldsymbol{U}$ is a solution of Eq. (2.68). The advantage of the weak solution is that it accommodates discontinuous solutions that arises naturally in the solution of the hyperbolic Euler equation system. When the weak problem is integrated over a bounded space in $x-t$ plane with the presence of a shock the integral weak form reproduce the Rankine-Hugoniot conditions across the shock curve [LeVeque, 2002; Salsa, 2016; Michel-Dansac, 2016].

Vila [Vila, 1999] defined the weak formulation associated to Eq. (2.54) by

$$
\forall\varphi\in C^2_0\left(\mathbb{R}^d\times\mathbb{R}^{+,*}\right)\ \int_{\mathbb{R}^d\times\mathbb{R}^+}\left(\boldsymbol{U}L^*_{\boldsymbol{w}}(\varphi)+\boldsymbol{F}^\alpha_{\boldsymbol{w}}\frac{\partial\varphi}{\partial x^\alpha}+\boldsymbol{S}\varphi\right)dx^d\,dt=\boldsymbol{0}
\tag{2.74}
$$

The set of test functions $C^2_0\left(\mathbb{R}^n*\mathbb{R}^{+,*}\right)$ considered by Vila, verifies $\varphi(x,0)=0$ and so the term on the right hand side of Eq. (2.73) is dropped. So far we have considered

that the test function is a scalar and therefore all the three terms appearing inside brackets are the product of a vector $\in \mathbb{R}^p$ with a scalar. In this form the vector of conservative variables is given by a vector but the spatial gradients and divergence term appear with spatial index. The same result is obtained if we consider the test function as a vector $\boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p$ with all components equal to φ and replace the product of scalar with a vector with the element-wise product of two vectors. Using this vectorization of the test function Eq. (2.74) is rewritten as

$$\forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^d \times \mathbb{R}^+} \left(\boldsymbol{U}.L^*_{\boldsymbol{w}}(\boldsymbol{\varphi}) + \boldsymbol{F}^\alpha_{\boldsymbol{w}}.\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha} + \boldsymbol{S}.\boldsymbol{\varphi} \right) dx^d\, dt = \mathbf{0} \quad (2.75)$$

To summarize and remark some features we rewrite the continuous strong and weak forms together. Comparison of these two forms are of great importance to derive the discrete weak form and also to understand the connection with the introduction of the skew-adjoint operators.

$$\begin{aligned} S.F.: &\quad \forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^d \times \mathbb{R}^+} \left(L_{\boldsymbol{w}}(\boldsymbol{U}).\boldsymbol{\varphi} + \tfrac{\partial \boldsymbol{F}^\alpha_{\boldsymbol{w}}}{\partial x^\alpha}.\boldsymbol{\varphi} - \boldsymbol{S}.\boldsymbol{\varphi} \right) dx^d dt = \mathbf{0} \\ W.F.: &\quad \forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^d \times \mathbb{R}^+} \left(-\boldsymbol{U}.L^*_{\boldsymbol{w}}(\boldsymbol{\varphi}) - \boldsymbol{F}^\alpha_{\boldsymbol{w}}.\tfrac{\partial \boldsymbol{\varphi}}{\partial x^\alpha} - \boldsymbol{S}.\boldsymbol{\varphi} \right) dx^d dt = \mathbf{0} \end{aligned} \quad (2.76)$$

A comparison of the Strong Form and Weak Form given in Eq. (2.76) shows that both the integrands are composed of three terms. Jumping between these two forms relies on the compact support of the test functions and some continuous operators' properties, that were used while performing integration by parts. We note that the source term appears equal in both forms but the relation between the terms involving the transport operator and the flux divergence in the strong form are related with their counterparts in the weak form with opposite sign.

By comparison we can appreciate the following relations that are exact in the continuous level but that require special attention when particle approximation is introduced.

$$\int_{\mathbb{R}^d} \left(L_{\boldsymbol{w}}(\boldsymbol{U}).\boldsymbol{\varphi}\right) dx^d = -\int_{\mathbb{R}^d} \left(\boldsymbol{U}.L^*_{\boldsymbol{w}}(\boldsymbol{\varphi})\right) dx^d \quad (2.77)$$

$$\int_{\mathbb{R}^d} \left(\frac{\partial \boldsymbol{F}^\alpha_{\boldsymbol{w}}}{\partial x^\alpha}.\boldsymbol{\varphi} \right) dx^d = -\int_{\mathbb{R}^d} \left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}.\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha} \right) dx^d \quad (2.78)$$

Discrete weak form

In order to obtain a discrete version of Eq. (2.75), we start by decomposing the integral using Fubini's theorem, giving

$$\forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^+} \left[\int_{\mathbb{R}^d} \left(\boldsymbol{U}.L^*_{\boldsymbol{w}}(\boldsymbol{\varphi}) + \boldsymbol{F}^\alpha_{\boldsymbol{w}}.\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha} + \boldsymbol{S}.\boldsymbol{\varphi} \right) dx^d \right] dt = \mathbf{0} \quad (2.79)$$

We need now to approximate the integration over the space $\mathbb{R}^d$ by using the particle quadrature $\int \phi(x,t)dx^d \approx \sum \phi\left(x_j(t)\right) V_j(t)$. To accommodate the quadrature rule for the component-wise product of two vectors in $\mathbb{R}^p$, we provide the space with the discrete scalar product $(.,.)_\Delta$ as :

$$\int_{\mathbb{R}^d} \boldsymbol{\varphi}.\boldsymbol{\Psi} dx^d \longrightarrow (\boldsymbol{\varphi},\boldsymbol{\Psi})_\Delta := \sum_i V_i \boldsymbol{\varphi}_i.\boldsymbol{\Psi}_i = \int_{\mathbb{R}^d} \bar{\boldsymbol{\varphi}}^\Delta.\bar{\boldsymbol{\Psi}}^\Delta dx^d \tag{2.80}$$

where $\bar{\boldsymbol{\varphi}}^\Delta$ and $\bar{\boldsymbol{\Psi}}^\Delta$ are used to show that the quadrature formulae approximation would be exact for that piecewise constant vector fields.

After application of the particle quadrature given by Eq. (2.80) inside each term of the integrand of Eq. (2.79), we obtain

$$\forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^+} \left[\left(\boldsymbol{U}, L^*_{\boldsymbol{w}}(\boldsymbol{\varphi})\right)_\Delta + \left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}, \frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha}\right)_\Delta + \left(\boldsymbol{S}, \boldsymbol{\varphi}\right)_\Delta\right] dt = \mathbf{0} \tag{2.81}$$

Once the quadrature has been applied, the next step demands the introduction of an approximation operator for the derivative $\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha}$. According to Eq. (2.50) the approximation for the gradient that assures zero consistency is defined by

$$\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha} \longrightarrow D^\alpha_h(\boldsymbol{\varphi})(\boldsymbol{x}) = \mathit{\Pi}^h\left(\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha}\right)(\boldsymbol{x}) - \varphi(\boldsymbol{x})\mathit{\Pi}^h\left(\frac{\partial 1}{\partial x^\alpha}\right)(\boldsymbol{x}) \tag{2.82}$$

that far away from the boundary takes the final form

$$D^\alpha_h(\boldsymbol{\varphi})(\boldsymbol{x}) = \sum_j \left(\boldsymbol{\varphi}_j - \boldsymbol{\varphi}(\boldsymbol{x})\right) \frac{\partial W}{\partial x^\alpha}\left(\boldsymbol{x} - \boldsymbol{x}_j, h\right) V_j \tag{2.83}$$

With incorporation of this second approximation, Eq. (2.81) becomes

$$\forall \boldsymbol{\varphi} \in \left(C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^+} \left[\left(\boldsymbol{U}, L^*_{\boldsymbol{w}}(\boldsymbol{\varphi})\right)_\Delta + \left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}, D^\alpha_h(\boldsymbol{\varphi})\right)_\Delta + \left(\boldsymbol{S}, \boldsymbol{\varphi}\right)_\Delta\right] dt = \mathbf{0} \tag{2.84}$$

It seems now that the only way to proceed further is developing the scalar products and the approximation for the derivative. But instead, we impose the relations between the strong and weak form obtained in the continuous situation. When the particle quadrature given by Eq. (2.80) is applied in the continuous relations of Eq. (2.77) and Eq. (2.78), the following identities should be verified by the discrete operators:

$$\int_{\mathbb{R}^d} \left(L_{\boldsymbol{w}}(\boldsymbol{U}).\boldsymbol{\varphi}\right) dx^d = -\int_{\mathbb{R}^d} \left(\boldsymbol{U}.L^*_{\boldsymbol{w}}(\boldsymbol{\varphi})\right) dx^d \longrightarrow \left(L_{\boldsymbol{w}}(\boldsymbol{U}), \boldsymbol{\varphi}\right)_\Delta = -\left(\boldsymbol{U}, L^*_{\boldsymbol{w}}(\boldsymbol{\varphi})\right)_\Delta \tag{2.85}$$

$$\int_{\mathbb{R}^d} \left(\frac{\partial \boldsymbol{F}^\alpha_{\boldsymbol{w}}}{\partial x^\alpha}.\boldsymbol{\varphi}\right) dx^d = -\int_{\mathbb{R}^d} \left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}.\frac{\partial \boldsymbol{\varphi}}{\partial x^\alpha}\right) dx^d \longrightarrow \left(D^\alpha_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}}), \boldsymbol{\varphi}\right)_\Delta = -\left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}, D^{*\alpha}_h(\boldsymbol{\varphi})\right)_\Delta \tag{2.86}$$

It should be noted that in the case of Eq. (2.85), the definition of the operator $L^*_{\boldsymbol{w}}$ was introduced during the continuous weak form derivation, meanwhile in Eq. (2.86) the definition of $D^{*\alpha}_h(\boldsymbol{\varphi})(\boldsymbol{x})$ is introduced to force the discretized system to fulfill the relation given by Eq. (2.86). Identities are analogue with respect to the scalar product $(.,.)_\Delta$. It is noticed that $-L^*_{\boldsymbol{w}}$ is the adjoint operator of $L_{\boldsymbol{w}}$ and that $-D^*_h$ is the adjoint operator of D_h.

By using the relation given by Eq. (2.85) and the fact that $\left(D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}}), \boldsymbol{\varphi}\right)_\Delta = -\left(\boldsymbol{F}^\alpha_{\boldsymbol{w}}, D^\alpha_h(\boldsymbol{\varphi})\right)_\Delta$ (which is equivalent to Eq. (2.86)), the discrete weak formulation given by Eq. (2.84) reads as

$$\forall \boldsymbol{\varphi} \in \left(C^2_0\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^+} \Big[-\left(L_{\boldsymbol{w}}(\boldsymbol{U}), \boldsymbol{\varphi}\right)_\Delta - \left(D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}}), \boldsymbol{\varphi}\right)_\Delta + \left(\boldsymbol{S}, \boldsymbol{\varphi}\right)_\Delta \Big] dt = \mathbf{0} \tag{2.87}$$

Expanding the discrete scalar products and noting that all terms are multiplied by $\boldsymbol{\varphi_i}$, the weak formulation can be stated as

$$\forall \boldsymbol{\varphi} \in \left(C^2_0\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right)\right)^p \int_{\mathbb{R}^+} \left[\sum_i \Big(V_i L_{\boldsymbol{w}}(\boldsymbol{U}_i) + V_i D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i - V_i \boldsymbol{S}_i \Big) . \boldsymbol{\varphi_i} \right] dt = \mathbf{0} \tag{2.88}$$

Since this weak formulation must be verified for any test function with compact support, a proper selection of the support domain implies that, for all particle i

$$V_i L_{\boldsymbol{w}}(\boldsymbol{U}_i) + V_i D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i = V_i \boldsymbol{S}_i \tag{2.89}$$

The use of the Generalized Reynolds Transport (See Eq. (2.10)) allows us to identify the first term as $V_i L_{\boldsymbol{w}}(\boldsymbol{U}_i) = \frac{d}{dt}\left(\boldsymbol{U}_i V_i\right)$, and this result in the weak semi-discretized system for the Euler equations

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i = V_i \boldsymbol{S}_i \tag{2.90}$$

To close the weak derivation of the SPH-ALE method we summarize now the discretized form obtained for the Euler equations

$$\begin{cases} (i) & \frac{d}{dt}\boldsymbol{x}_i = \boldsymbol{w}\left(x_i, t\right) & \boldsymbol{x_i}(0) \\ (ii) & \frac{d}{dt} V_i = D^\alpha_h(w^\alpha_i) V_i & V_i(0) \\ (iii) & \frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i D^{*\alpha}_h(\boldsymbol{F}^\alpha_{\boldsymbol{w}})_i = V_i \boldsymbol{S}_i & \boldsymbol{U}_i(0) \end{cases} \tag{2.91}$$

where (i) tracks the position of particles according to the velocity field $\boldsymbol{w}$, (ii) updates the weight volume of particles starting from an initial volume weight V_i and (iii) expresses the conservation of mass, momentum and energy in ALE conservative form.

It is interesting to remark that the procedure naturally prescribes different gradient approximations for the divergence of the velocity and for the divergence of the ALE fluxes. This is a notorious difference with the strong form derived in Eq. (2.64). The weak form formulation imposes a discretization of the convective fluxes that assures

the compatibility between the continuous and discrete operators. Compatibility is a desirable property for a meshless formulation because it forces that integration by parts is satisfied at discrete level [Fougeron, 2018]. The concept of compatibility is not exclusive of SPH-ALE methods and it has also been invoked to improve traditional Lagrangian SPH methods ([Kulasegaram et al., 2004], [Mayrhofer et al., 2013]), other meshless formulations ([Trask et al., 2017]) and even mesh-based methods ([Lipnikov & Shashkov, 2010]).

2.5. Utilization of Riemann solvers in SPH-ALE

Some of the properties that need to be improved in SPH methods are convergence, consistency and stability [Vacondio et al., 2020]. In this framework, the use of Riemann solvers is a promising option to increase the stability of the numerical methods. In particular, this work is based on the SPH-ALE method [Vila, 1999; Ben Moussa & Vila, 2000]. In this scheme, Riemann solvers are used instead of artificial dissipation to stabilize the method. The Riemann problem is solved between two neighboring particles on the direction of the line connecting them. Left and right Riemann states are defined using the values of the variables on each of the neighboring particles, and Taylor series expansions of the variables at integration points are used to improve the accuracy of the SPH scheme.

The SPH-ALE formulation was introduced by Vila and Ben Moussa [Ben Moussa & Vila, 2000; Vila, 1999] to increase the accuracy and stability of SPH methods in nonlinear systems of conservation laws. Vila and Ben Moussa applied this formulation to the Euler equations and presented a system of equations in semidiscrete form that has many similarities with the finite volume formalism.

In the SPH-ALE formulation, the interaction of each neighboring particle j with the particle i admits a representation as a flux at the midpoint ij located at $\boldsymbol{x}_{ij} = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_j)$. Fluxes are computed from solutions to one-dimensional moving Riemann problems. Thus, we can associate the particle i as the left state, particle j as the right state and the moving interface with the midpoint ij. Figure 2.12 shows the definition of one of the moving Riemann problems. Unit vector $\boldsymbol{n}_{ij}$ points from particle i to particle j. We use index ij^- and ij^+ to denote reconstruction values at the interface from the left and from the right. The kernel gradient can be expressed in terms of the unit vector $\boldsymbol{n}_{ij}$. Kernel functions which depend only on the distance between particles can be expressed as $W_{ij} = W(\boldsymbol{x}_i - \boldsymbol{x}_j, h_{ij}) = W(q_{ij})$, where $q_{ij} = \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{h_{ij}}$.

The gradient of the kernel function is given by $\nabla W_{ij} = |\frac{\partial W}{\partial q_{ij}}|\frac{1}{h_{ij}}\boldsymbol{n}_{ij}$ showing that $\boldsymbol{n}_{ij}$ and ∇W_{ij} are vectors with the same direction.

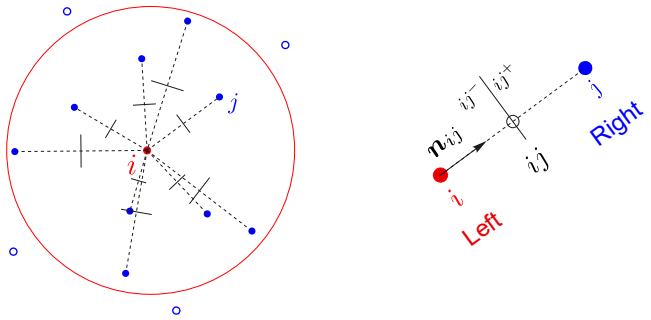


Figure 2.12. Accommodation of Riemann solvers in SPH-ALE. On the left: Stencil of neighboring particles for a particle P_i. The neighboring particles inside the kernel support of P_i are represented with blue filled circles. Dotted lines connect the position of P_i with its interacting neighbors. On the right: An enlarged view of P_i with one of its interacting neighbors P_j. The interaction between particles P_i and P_j is accounted as the the flux in the midpoint interface ij of a one-dimensional moving Riemann problem. The state of P_i and P_j are associated with the left and right states.

2.6. Implementation of boundary conditions

Implementation of boundary conditions is recognized as one of the main challenges that SPH and meshless methods have to face. Although the free surface condition can be easily handled by Lagrangian meshless methods, the imposition of other types of boundary conditions is more cumbersome in meshless than in mesh-based methods. Due to the Lagrangian nature of SPH, imposing boundary condition types in stationary regions is not a trivial task. For instance, the simple imposition of a perfect impermeable wall is not easily enforced in a traditional Lagrangian SPH formulation. It should be noted that the Lagrangian SPH formulation was conceived as a numerical method to solve partial differential equations in an infinity domain. When the method is used to solve problems in bounded domains it becomes necessary to impose the common type of boundary conditions.

In traditional SPH methods, different strategies to implement boundary conditions has been considered. In the case of wall boundary type, the different strategies adopted in SPH community can be classified in four groups: boundary repulsive force [Monaghan, 1994; Monaghan & Kajtar, 2009], the mirror particles [Morris et al., 1997; Szewc, 2013], dynamic boundary particles [Crespo et al., 2007; Marrone et al., 2011; Adami et al., 2012a; English et al., 2021] and semi-analytical boundary conditions [Mayrhofer et al., 2013; Ferrand et al., 2017].

The same strategies to impose boundary conditions in traditional SPH methods can be implemented in SPH-ALE framework. However, the similarity of SPH-ALE with Finite Volume Methods and the utilization of Riemann solvers, enable to consider different techniques. Marongiu [Marongiu, 2007] applied the theory of partial Riemann solvers [Dubois, 2001] to implement boundary conditions in a weakly com-

pressible SPH-ALE code. Li et Marongiu [Li, 2013; Li et al., 2014] also employ the partial Riemann technique to couple a weakly compressible SPH-ALE fluid solver with a Total Lagrangian SPH solver for the solid. Neuhauser et al. [Neuhauser, 2014] implemented boundary conditions in a SPH-ALE solver according to the Characteristic Boundary Conditions proposed by Poinsot [Poinsot & Lele, 1992]. Werdelmann et al. [Werdelmann et al., 2021] provided the procedure to implement the Characteristic Boundary Conditions in SPH method expressed in ALE framework.

In this thesis the mirror technique was employed to implement boundary conditions in the SPH-ALE method presented in Chapter 3. Details about the implementation of this technique in SPH-ALE formulation are provided in the test cases considered. The main contributions to the imposition of boundary conditions is associated with the MLSPH-ALE formulation presented in Chapter 4. The MLSPH-ALE method provides an additional boundary term that eases the implementation of boundary conditions. The imposition of the no-slip wall in the flow around a circular cylinder was conducted using this novel technique.

2.7. Conclusions

In this chapter the fundamentals of the kernel approximation used by SPH method were exposed. Kernel approximation involves a two-step process. Starting with the identity given by the convolution formula, the first step in the approximation replaces the Dirac function by a kernel function, and the second step approximates the integral by a summation over a set of neighbor particles. Although the integral kernel approximation achieves second order accuracy, the disordered position of neighbor particles degrades the accuracy of the discrete kernel approximation.

Different derivations of the SPH-ALE meshless formulation were detailed. SPH-ALE method comes with some important advantages over traditional SPH formulations. Apart from the flexibility given by the ALE framework another important advantage of this meshless formulation is the interpretation of interaction between particles as a Riemann flux. The semi-discrete system obtained by SPH-ALE method has great similarities with the semi-discrete system obtained with mesh-based methods. This analogy between meshless and mesh-based methods suggest the possibility of explore enhancements in one type of methods by the transfer of successful techniques in the other type.

Chapter 3

A Weakly Compressible SPH-ALE method

3.1. Introduction

In this chapter a high accuracy meshless method is proposed to solve liquid flows. The method is based on the meshless framework proposed by Vila and Ben Moussa [Vila, 1999; Ben Moussa & Vila, 2000]. Meshless methods based on Vila's idea are obtained by invoking a local weak Galerkin formulation over a conservation law in ALE description using the SPH kernel as test and trial functions. Meshless methods based on this formulation are usually labeled as SPH-ALE methods since they retain the SPH kernel approximation, and differ from the traditional Lagrangian methods in the ALE capability. In SPH-ALE methods the interaction between neighbor particles can be interpreted as flux transfer terms, in the same manner that is done in mesh-based Finite Volume Methods. Indeed, in Astrophysics community SPH-ALE methods are usually referred by meshless finite volume methods [Ivanova et al., 2013; Hopkins, 2015]. Vila also showed that minor realignment of terms enables SPH-ALE methods to incorporate approximate Riemann solvers for the numerical approximation of hyperbolic flux terms. SPH-ALE methods can overcome some of the problematic issues of Lagrangian SPH methods like the noisy pressure field or the undesirable particle pattern positions [Oger et al., 2016].

In this chapter, an improved SPH-ALE method to solve liquid flows is presented. The new method recognizes the inheritance from Vila framework and appends Moving Least Squares (MLS) and Multidimensional Optimal Order Detection (MOOD) techniques to increase its accuracy and keep stability, respectively. It is now time to explain some concepts about the different approaches to model liquid flows.

Liquids are fluids with low compressibility coefficient, but this does not imply that compressible effects can be systematically neglected. Sonar applications, underwater

explosions or water hammer in pipe networks are applications where the compressibility effects are relevant. In liquid flow with negligible compressibility effects, the selection of an equation of state (EOS) with constant density is appropriate. On the contrary, for liquid flows with non-negligible compressibility, it is mandatory to use an EOS that reflects the relationship between the thermodynamic properties. The selected EOS must reproduce the real properties of the substance in order to respect its thermodynamic equilibrium state.

We can follow an incompressible or compressible strategy to solve the governing equations. The incompressible approach assumes that the density is constant, and thus the continuity equation imposes a solenoidal field for the velocity field. The compressible approach solves the full set of equations, with the EOS providing a link between the density, pressure and temperature.

Thus, we have two types of models for the liquid properties and two types of approaches for the numerical solution of the governing equations. According to the physics, it seems reasonable to select the approach for the numerical solution that corresponds with the importance of fluid compressibility effects in a particular problem. However, both approaches result in different character of the numerical equations to solve, and thus different computational requirements. The incompressible approach renders the mathematical problem into a large system of linear equations, meanwhile the compressible approach results in an explicit scheme that only needs to evaluate expressions to advance the solution locally in time. When numerical and computational issues are taking into account, the selection of the best approach for a certain application is a question that generates great debate in literature. Focusing on liquid flows with negligible compressibility effects ($\rho \approx constant$), it is possible to adopt the incompressible or compressible approach. As the compressibility effects are negligible the compressible approach does not require to use the real EOS of the fluid and it can use a modified version of the EOS with a reduced sound propagation speed, that makes this approach less demanding by alleviating the CFL time constraint. Thus, the weakly compressible approach uses the compressible approach for flows with negligible compressibility effects.

Similar to the Finite Volume Method [Ramírez et al., 2014; Nogueira et al., 2016b], in SPH there are two main approaches to model liquids. One is based on the incompressibility assumption of the Navier-Stokes equations. This assumption leads to the decoupling of the equations and the continuity equation can be considered as a constraint the velocity field has to satisfy. The methods are based on the solution of a Poisson equation for the pressure field, using the pressure-correction idea from grid-based methods [Cummins & Rudman, 1999; Garoosi & Shakibaeinia, 2020] . This approach is known as Incompressible SPH (ISPH). The second approach, introduced by Monaghan [Monaghan, 1994] is based on Weakly Compressible hypotheses (WCSPH). In this approach, the incompressibility is approximated by artificially allowing a slight flow compressibility. One advantage of this approach is that it avoids the need for solving a Poisson equation to compute the pressure field. The computation

of pressure only requires the use of an equation of state. As the density of most liquids is nearly constant, a barotropic approximation is reasonable, and a linear EOS depending only on density is often used [Collé et al., 2019]. Both approaches have advantages and drawbacks. Thus, one advantage of weakly-compressible methods, is that these schemes are more suited for free-surface flows as the boundary condition along the free surface is implicitly satisfied, and do not require an explicit detection of the free surface during the flow evolution. ISPH schemes are more difficult to parallelize because of the need for solving an algebraic system with a sparse matrix. However, the weakly-compressible approach requires small time steps (as it is constrained by the speed of sound), whereas ISPH allows for larger time steps. On the other hand, in weakly compressible approach, oscillations in density and pressure typically appear in the solution. In order to alleviate these oscillations, several authors have proposed two different procedures. The first one was introduced by Colagrossi et al. [Colagrossi et al., 2012], proposing a filtering of the density field. It reduces the numerical noise by restoring the consistency between mass, density and volume. The second procedure is more recent, and was introduced by Marrone et al. [Marrone et al., 2011]. They developed the δ-SPH scheme, in which a density diffusive term is added to smooth the spurious density oscillations.

This chapter deals with SPH models based on the weakly compressible approach to solve liquid flows. A wide set of test cases are reproduced to analyze the performance of the proposed SPH-ALE-MOOD. The weakly compressible approach can be considered as a particular case of the compressible approach with an EOS adjusted conveniently. Based on that fact some compressible liquid flows were included as validation cases.

The structure of the chapter starts with the presentation of the governing equations for compressible flows. Then the most used constitutive equations for compressible liquids are presented. After that the traditional discrete equations of the Lagrangian SPH and the SPH-ALE method are presented. Next subsection enumerates some works in the literature concerning SPH-ALE scheme. Then the improved SPH-ALE-MOOD method is presented providing details of its new novelties. In the end of the chapter the method is applied over a set of test cases.

3.2. Governing equations

Adopting an ALE approach the Navier-Stokes system of equations can be expressed in a differential conservative form by

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \nabla \cdot (\boldsymbol{F}_E - \boldsymbol{w} \otimes \boldsymbol{U}) - \nabla \cdot (\boldsymbol{D}) = \boldsymbol{S} \tag{3.1}$$

where $\boldsymbol{w}$ stands for a regular velocity field and $\boldsymbol{U}$ is the vector of conservative variables. The operator $L_{\boldsymbol{w}}$ is called the transport operator linked to $\boldsymbol{w}$. The application of this operator over $\boldsymbol{U}$ is designated by $L_{\boldsymbol{w}}(\boldsymbol{U})$ and corresponds to $L_{\boldsymbol{w}}(\boldsymbol{U}) = \partial_t \boldsymbol{U} + \nabla \cdot (\boldsymbol{w} \otimes \boldsymbol{U})$. The definition of the transport operator can be considered as the differential version of the Reynolds Transport Theorem applied over a control volume whose boundary

move with a velocity $\boldsymbol{w}$ (see Appendix B). We denote with $\boldsymbol{F_E}$ the Eulerian flux tensor (convective and pressure terms), $\boldsymbol{D}$ represents the viscous tensor and vector $\boldsymbol{S}$ contains the source terms.

For two dimensional cases the vectors and tensors previous introduced are given by

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \qquad \boldsymbol{F}_E^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho H u \end{pmatrix}, \qquad \boldsymbol{F}_E^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho H v \end{pmatrix} \tag{3.2}$$

$$\boldsymbol{D}^x = \begin{pmatrix} 0 \\ \tau^{xx} \\ \tau^{yx} \\ \tau^{xx}u + \tau^{xy}v - q^x \end{pmatrix}, \quad \boldsymbol{D}^y = \begin{pmatrix} 0 \\ \tau^{xy} \\ \tau^{yy} \\ \tau^{xy}u + \tau^{yy}v - q^y \end{pmatrix}, \tag{3.3}$$

$$\boldsymbol{S} = \begin{pmatrix} 0 \\ \rho f^x \\ \rho f^y \\ \rho f^x u + \rho f^y v + \dot{q_h} \end{pmatrix} \tag{3.4}$$

where the vectors fluxes $\boldsymbol{F}_E^x$ and $\boldsymbol{F}_E^y$ are the rows of the flux tensor $\boldsymbol{F}_E$, namely $\boldsymbol{F}_E = (\boldsymbol{F}_E^x, \boldsymbol{F}_E^y)^T$. Similarly the viscous tensor $\boldsymbol{D}$ obeys $\boldsymbol{D} = (\boldsymbol{D}^x, \boldsymbol{D}^y)^T$. Equations (3.2), (3.3) and (3.4) correspond with the expressions given by Eq (2.19) in Chapter 2 particularized for a two-dimensional domain.

The vector fluid velocity and its components in x and y direction are denoted by $\boldsymbol{u} = (u, v)^T$. Density and pressure are designed by ρ and p. We use E for the specific total energy defined as the sum of the internal energy (e) and the kinetic energy (ke) according to $E = e + \frac{1}{2}(u^2 + v^2)$. The total enthalpy definition is defined as $H = E + p/\rho$. For the diffusive terms $\tau^{\alpha\beta}$ denotes the viscous tensor component and q^α the thermal conduction flux component. For an incompressible Newtonian fluid $\tau^{\alpha\beta}$ can be expressed as $\tau^{\alpha\beta} = \mu(\frac{\partial u^\alpha}{\partial x^\beta} + \frac{\partial u^\beta}{\partial x^\alpha})$ where μ is the dynamic fluid viscosity. Similarly, the thermal flux is expressed in terms of temperature gradients and thermal conductivity, according to $q^\alpha = -\lambda(\frac{\partial T}{\partial x^\alpha})$. Finally, the vector $\boldsymbol{f} = (f^x, f^y)^T$ represent external force components per unit mass, and $\dot{q}_h$ is a volumetric heat source.

3.3. Constitutive Equations for compressible liquids

Two different equations of state (EOS) for compressible liquids are chosen: Tait and Tammann EOS but some other EOS for liquids are available in the literature [Denner, 2021; Chandran & Salih, 2019].

Tait EOS models a barotropic fluid, and the pressure only depends on the density, that is, $p = p(\rho)$. Tammann EOS is more general and relates pressure with both the density and the internal energy, that is, $p = p(\rho, e)$. Tait EOS keeps the energy

equation decoupled from the momentum equation and can lead to computational cost savings when energy effects on the flow are negligible. However, when shock waves are present in the flow, the Tammann EOS is a more convenient choice.

Table 3.1 shows the expressions to evaluate pressure and acoustic sound speed for any of the EOS adopted in this work. Caloric equations are also provided although its inclusion is optional for a barotropic fluid. Expressions for Tait EOS were extracted from [Saurel et al., 1999] whereas for the Tammann stiffened gas EOS were extracted from [Paillère et al., 2003]. The last row in the table presents the whole set of constants values required to properly set each EOS. These values are fixed case by case in the validation section. Zero subindex in Tait equation means the constant is associated to the reference state of the fluid.

Tait EOS [Saurel et al., 1999]	Tammann EOS [Paillère et al., 2003]
$p(\rho) = \frac{\rho_0 c_0^2}{\gamma}\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) + p_0$	$p(\rho, e) = (\gamma - 1)\rho e - \gamma p_c$
$c(\rho) = c_0\left(\frac{\rho}{\rho_0}\right)^{\frac{\gamma-1}{2}}$	$c(\rho, e) = \sqrt{\frac{\gamma(p + p_c)}{\rho}}$
$e = c_v(T - T_0)$	$e = \frac{c_p}{\gamma}T + \frac{p_c}{\gamma}$
$\rho_0, c_0, \gamma, p_0, c_v$	γ, p_c, c_p

Table 3.1. Constitutive equations for the barotropic Tait EOS and for the Tammann stiffened gas EOS.

3.4. Discrete representation of the geometric domain

In this section we address the discretization of the computational domain. In this stage of the simulation process meshless and mesh-based methods separate their paths. In meshless methods the discretized geometry results in a cloud of points meanwhile in mesh-based methods the discretized geometry comprises a mesh or a grid.

Figure 3.1 depicts the reduction of a continuous geometric domain into a discrete one. The geometric domain Ω with boundary $\partial\Omega$ is discretized into a set of N computational points at positions $\boldsymbol{x}_i = (x_i, y_i)^T$ expressed with respect to a Cartesian coordinate system. Index i is used to label computational points and ranges from 1 to N. Moreover, each particle i has n_i interacting neighboring particles inside its compact support domain D_i with boundary ∂D_i.

Each computational point has properties and associated values of the variables that evolve in time. The set of data linked to computational points depends on the type of meshless method. Traditional Lagrangian SPH and SPH-ALE methods differ in the set of properties attached to particles. For any generic scalar variable ϕ and vector variable $\boldsymbol{\phi}$ the quantity carried by computational point i is denoted by ϕ_i and $\boldsymbol{\phi}_i = (\phi_i^1, ..., \phi_i^d)^T$ respectively.

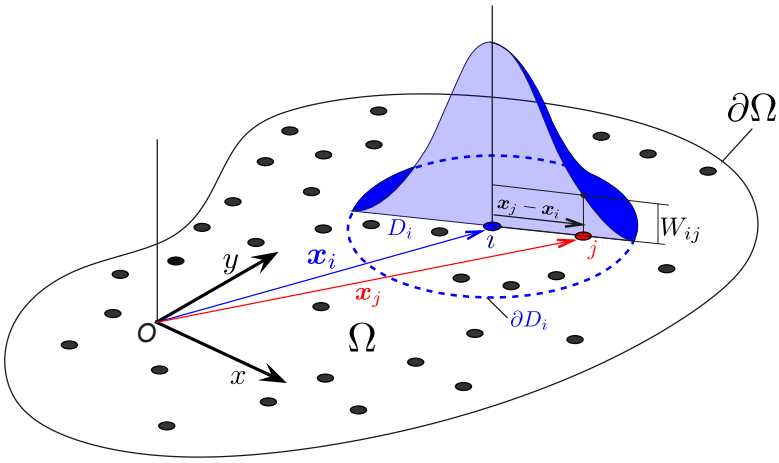


Figure 3.1. Computational domain Ω and kernel support D_i of particle i.

3.5. Some improvements for the SPH-ALE method

Several authors have proposed the SPH-ALE formulation for the Navier–Stokes equations [Chiron, 2017; Oger, 2006; Sjah, 2013]. In these works, the viscous term was discretized using an approximation of the Laplacian based on a hybrid SPH gradient by means of a first-order finite difference scheme [Morris, 1996]. In this work, we propose a different discretization for the viscous term of the Navier-Stokes equations. Observation of the Navier-Stokes in ALE form given by Eq. (3.1) suggests that the viscous terms can be computed in the form of a diffusive flux, following a similar approach as the one used for the convective terms.

Thus, the proposed resulting semi-discretized form of the Navier-Stokes equations is given by

$$\begin{aligned}\frac{d(V_i \boldsymbol{U}_i)}{dt} = &- \sum_{j=1}^{n_i} V_i V_j 2 \left(\boldsymbol{G}_{ij} - (\boldsymbol{F}_{\boldsymbol{w}})_i\right) \cdot \nabla W_{ij} \\ &+ \sum_{j=1}^{n_i} V_i V_j 2 \left(\boldsymbol{D}_{ij} - \boldsymbol{D}_i\right) \cdot \nabla W_{ij} + V_i \boldsymbol{S}_i\end{aligned} \tag{3.5}$$

$$\frac{dV_i}{dt} = \sum_{j=1}^{n_i} V_i V_j 2 \left(\boldsymbol{w}_{ij} - \boldsymbol{w}_i\right) \cdot \boldsymbol{\nabla} W_{ij} \tag{3.6}$$

where Eq. (3.5) expresses the evolution of the conservative variables and Eq. (3.6) describes the evolution of the effective volume associated to particle P_i. In the above equations, $W_{ij} = W(\boldsymbol{x}_i - \boldsymbol{x}_j, h_{ij})$ is a kernel function and $h_{ij} = \frac{1}{2}(h_i + h_j)$ is the averaged smoothing length. The smoothing length h_i of a particle P_i is linked to its volume V_i via equation $h_i = \beta V_i^{\frac{1}{d}}$, where β is a constant and d is the dimension of the

computational domain. Note that, as the interparticle distance dx can be estimated as $dx = V_i^{\frac{1}{d}}$, we adopt the practical consideration of linking the smoothing kernel length by a constant factor β to the interparticle distance dx.

The support radius for a kernel function is expressed as the product of a parameter κ and the smoothing length h. For a computational particle P_i it is possible to express the radius of its kernel support as $R_i = \kappa h_i$. Using the relation $h_i = \beta V_i^{\frac{1}{d}}$ we obtain that the radius of the kernel support for particle P_i is given by $R_i = \kappa\beta V_i^{\frac{1}{d}}$. For the cubic spline kernel used in this thesis $\kappa = 2$ and the radius of the kernel is given by $R_i = 2\beta V_i^{\frac{1}{d}}$. Since the initial volume of the particle is imposed by the initial conditions and Eq. (3.6) gives its evolution, the parameter β lets to modify the size of the kernel support and thus the number of interacting neighbors. Throughout this work, we have set $\beta = 2$. This implies that, for an initial uniform distribution of particles, $R = 4dx$, resulting in 9 neighbors for 1D tests and 49 neighbors for 2D problems.

Tensors $\boldsymbol{G}_{ij}$ and $\boldsymbol{D}_{ij}$ in Eq. (3.5) account for the inviscid and viscous fluxes in the interface ij, respectively. The terms appearing with minus sign inside the parenthesis ($(\boldsymbol{F_w})_i$ and $\boldsymbol{D}_i$) are tensors evaluated at the position of particle i that assure at least zero order consistency at discrete level as indicated by Avesani et al. [Avesani et al., 2014]. The origin of the term $(\boldsymbol{F_w})_i$ will be further discussed in the next chapter.

In the SPH-ALE formulation, each particle i is associated with a velocity frame $\boldsymbol{w}_i$ and a material velocity $\boldsymbol{u}_i$. The velocity frame $\boldsymbol{w}_i$ can be freely chosen and determines the evolution of particle positions. For the Eulerian approach of the method we set $\boldsymbol{w}_i = 0$ and particles are fixed in space. For the Lagrangian version, we set $\boldsymbol{w}_i = \boldsymbol{u}_i$ and perform a weighted average interpolation of the velocity [Bonet & Lok, 1999] to update particle positions. Therefore, the evolution of the particle position must satisfy for Eulerian/Lagrangian frame Eq. (3.7).

$$\frac{d\boldsymbol{x}_i}{dt} = 0 \qquad or \qquad \frac{d\boldsymbol{x}_i}{dt} = \frac{\sum_{j=1}^{n_i} V_j \boldsymbol{w}_j W_{ij}}{\sum_{j=1}^{n_i} V_j W_{ij}} \tag{3.7}$$

3.6. High-order discretization of the numerical fluxes

3.6.1. Moving Least Squares Approximation

Moving Least Squares is an approximation technique widely used in meshless methods with different purposes. In standard SPH methodology MLS is used as an auxiliary tool for boundary treatment [Marrone et al., 2011], field refreshing to remedy the noisy pressure field [Chen et al., 2013], reordering SPH methods or simply as an approximation technique in the post-processing stage. In SPH-ALE methods [Avesani et al., 2014; Nogueira et al., 2016a] MLS technique is used to increase the accuracy of convective fluxes.

First, we give a brief overview of the Moving Least Squares (MLS) technique for introducing the notations. We refer the reader to [Lancaster & Salkauskas, 1981] for a complete description of the technique and to [Chen et al., 2017] for their application in the context of meshless methods.

The MLS approximation of a function u at point $\boldsymbol{x} = (x, y, z)^T$ is approximated in the neighborhood around a particle i by a set of n_i values, u_j, and it is defined as

$$\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{n_i} N_j(\boldsymbol{x}) u_j \tag{3.8}$$

where the associated shape functions, gathered in vector $\boldsymbol{N} = (N_1, N_2, ..., N_{n_i}) \in \mathbb{R}^{n_i}$, are computed by

$$\boldsymbol{N}^T(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x}) \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{P}(\boldsymbol{x}) \boldsymbol{W}_{MLS}(\boldsymbol{x}) \tag{3.9}$$

where $\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y, z, x^2, y^2, z^2, xy, xz, yz...) \in \mathbb{R}^m$ is a m-dimensional basis functions vector, $\boldsymbol{P}(\boldsymbol{x})$ is a $m \times n_i$ matrix where the basis functions are evaluated at each point of the stencil of the particle i (namely $\boldsymbol{P} = [\boldsymbol{p}^T(\boldsymbol{x_j})]_i$) and $\boldsymbol{M}(\boldsymbol{x})$ is the $m \times m$ moment matrix given by

$$\boldsymbol{M}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}) \boldsymbol{W}_{MLS}(\boldsymbol{x}) \boldsymbol{P}^T(\boldsymbol{x}) \tag{3.10}$$

Diagonal matrix $\boldsymbol{W}_{MLS}(\boldsymbol{x})$ is derived from the kernel function evaluated at $\boldsymbol{x}_j - \boldsymbol{x}_i$ for the n_i neighboring particles [Cueto-Felgueroso et al., 2007].

Note that the general MLS technique allows that $\boldsymbol{x}$ can take any value in the domain Ω resulting in a continuous approximation. Therefore, the approximation of the variables or its derivatives can be obtained at any point of the domain. In Figure 3.2 a schematic representation of the MLS reconstruction is shown.

In the following sections we address the discretization of the convective and diffusive fluxes.

3.6.2. Discretization of the non-viscous flux

The numerical flux tensor $\boldsymbol{G}_{ij}$ is computed using the Rusanov flux in the co-moving frame according to

$$\boldsymbol{G}_{ij} = \frac{1}{2}\left[(\boldsymbol{F}_{\boldsymbol{w}})^+_{ij} + (\boldsymbol{F}_{\boldsymbol{w}})^-_{ij}\right] - \frac{1}{2} S^+_{ij} \Delta \boldsymbol{U}_{ij} \cdot \boldsymbol{n}$$

where $(\boldsymbol{F}_{\boldsymbol{w}})^-_{ij} = \boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{U}^-_{ij}, \boldsymbol{w}_{ij})$ and $(\boldsymbol{F}_{\boldsymbol{w}})^+_{ij} = \boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{U}^+_{ij}, \boldsymbol{w}_{ij})$ denote the approximations of the Lagrangian flux tensor $\boldsymbol{F}_{\boldsymbol{w}} = \boldsymbol{F}_E(\boldsymbol{U}) - \boldsymbol{w} \otimes \boldsymbol{U}$ on the left and right sides of the interface ij.

The term S^+_{ij} is the maximum eigenvalue of the Jacobian matrix which in the Arbitrarian Lagrangian–Eulerian (ALE) framework reads

$$S^+_{ij} = \max\left[(\boldsymbol{u}^+_{ij} - \boldsymbol{w}_{ij}) \cdot \boldsymbol{n}_{ij} + c^+_{ij}, (\boldsymbol{u}^-_{ij} - \boldsymbol{w}_{ij}) \cdot \boldsymbol{n}_{ij} - c^-_{ij}\right]$$

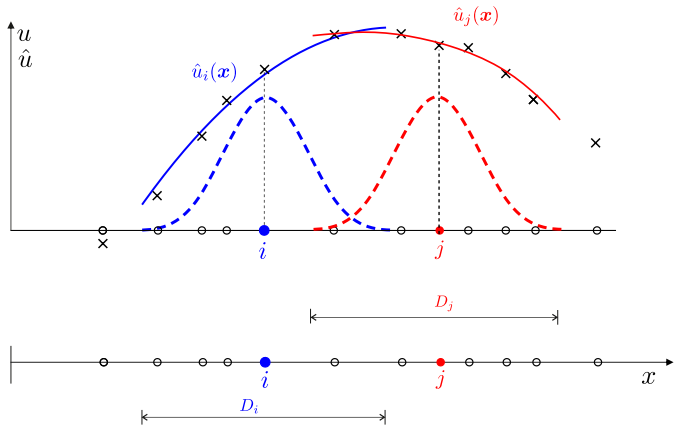


Figure 3.2. Schematic representation of MLS approximations. The blue solid line represents the MLS reconstruction associated to particle i, built from information of its neighboring particles. The blue dashed line represents the weight function. The same is plotted in red for another particle j. D_i and D_j are the supports of particles i and j.

where $\Delta \boldsymbol{U}_{ij} = \boldsymbol{U}_{ij}^{+} - \boldsymbol{U}_{ij}^{-}$ is the jump of the reconstructed conservative variables. Moreover, the term $\boldsymbol{w}_{ij}$ is the velocity of the reference frame at the interface ij. On an Eulerian frame, $\boldsymbol{w}_{ij} = 0$, whereas on a Lagrangian frame $\boldsymbol{w}_{ij} = \boldsymbol{u}_{ij} = (\boldsymbol{u}_{ij}^{+} + \boldsymbol{u}_{ij}^{-})/2$.

We note that, despite the known diffusive behavior of the Rusanov flux [Gallouët et al., 2002], it can be easily used with different EOS, so it is a convenient choice for the problems addressed here.

Tensor $(\boldsymbol{F}_{\boldsymbol{w}})_i = \boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{U}_i, \boldsymbol{w}_i)$ is the Lagrangian flux computed as a function of the state of the $i-$th particle $(\boldsymbol{F}_{\boldsymbol{w}})_i = \boldsymbol{F}_E(\boldsymbol{U}_i) - \boldsymbol{w}_i \otimes \boldsymbol{U}_i$.

One way to increase the accuracy of the resulting scheme is to compute the reconstruction of the variables at each integration point ij using a high-order approximation. For a given variable ϕ, which is known on each particle, we can compute the reconstructed variable at integration point, ϕ_{ij}, by means of Taylor series as

$$\phi_{ij}^{+} = \phi_i + \boldsymbol{\nabla}\phi_i \cdot (\boldsymbol{x}_{ij} - \boldsymbol{x}_i) + \frac{1}{2} (\boldsymbol{x}_{ij} - \boldsymbol{x}_i)^T \boldsymbol{\nabla}^2 \phi_i \, (\boldsymbol{x}_{ij} - \boldsymbol{x}_i) \tag{3.11}$$

where the first and successive derivatives are computed, following the previous work of Nogueira et al. [2016a], using MLS approximations.

3.6.3. *A posteriori* stabilization using the Multidimensional Optimal Order Detection (MOOD)

The *a posteriori* MOOD paradigm, introduced by Clain et al. [Clain et al., 2011], is used in this work to determine the optimal order of the polynomial reconstruction. This is determined iteratively by building a candidate solution $\mathbf{U}^*$ for time t^{n+1} based on

the t^n solution. The candidate solution is then tested through a series of detectors that check if the solution has a certain set of desirable properties. If any of the particles is flagged as invalid, the candidate solution at that particle is discarded and recomputed from the original solution at t^n but using a more dissipative scheme by lowering the polynomial reconstruction degree.

MOOD Loop The MOOD approach is composed of a Particle Polynomial Degree (PPD) and a chain of detectors. The PPD is associated to each particle i, and it refers to the actual polynomial degree used to compute the candidate solution $\mathbf{U}^*$ at particle i. We evaluate the flux at the midpoint ij between particles i and j, taking the minimum of the respective PPDs for the polynomial reconstruction as $PPD_{ij} = \min\left(\mathrm{PPD}(i), \mathrm{PPD}(j)\right)$. The chain of detectors controls the validity of the resulting solution, and the PPD is decremented where any of the detectors flag the solution as invalid.

The MOOD loop iterates through the PPD map, initialized with maximal order (in this case, $d_{\max} = 3$), decreasing the order of the particles that present a non-physical or invalid solution (that is, a solution that was flagged as invalid by the chain of detectors). In this work only third and first order schemes are used. The first order scheme is called the*parachute scheme* that, by definition, fulfills all the detectors requirements.

Chain Detectors In order to obtain a stable solution within the SPH formulation, a chain of detectors is used to assess whether the solution is admissible or not. In this work, we employ two detectors:

> **Physical Admissibility Detector (PAD)**: it checks that all the particles in the solution have positive density and pressure at all times. It also accounts for `NaN` (Not a Number) values that arise in the candidate solution.
>
> **Numerical Admissible Detector (NAD)** [Dumbser et al., 2014]: relaxed version of the Discrete Maximum Principle (DMP)[Clain et al., 2011]. It checks that the solution is monotonic and thus, no new extrema are created. It compares the candidate solution with the solution obtained in the previous Runge-Kutta step.

$$\min_{\mathbf{y}\in\mathcal{V}_i}\left(\mathbf{U}^{\mathrm{RK}}\left(\mathbf{y}\right)\right) - \delta \leqslant \mathbf{U}^*(\mathbf{x}) \leqslant \max_{\mathbf{y}\in\mathcal{V}_i}\left(\mathbf{U}^{\mathrm{RK}}\left(\mathbf{y}\right)\right) + \delta \tag{3.12}$$

where $\mathcal{V}_i$ is the set of closest particles and the tolerance δ is defined following [Dumbser et al., 2014] as

$$\delta = \max\left(10^{-4}, 10^{-3}\cdot\left(\max_{\mathbf{y}\in\mathcal{V}_i}\left(\mathbf{U}^{\mathrm{RK}}\left(\mathbf{y}\right)\right) - \min_{\mathbf{y}\in\mathcal{V}_i}\left(\mathbf{U}^{\mathrm{RK}}\left(\mathbf{y}\right)\right)\right)\right) \tag{3.13}$$

3.6.4. Discretization of the viscous flux

In this work we propose to extend the same discretization that is typically performed in the Finite Volume method to SPH-ALE discretizations. Tensor $\boldsymbol{D}_i = \boldsymbol{D}(\boldsymbol{U}_i, \boldsymbol{\nabla}\widehat{\boldsymbol{U}}_{\boldsymbol{i}})$ is the viscous tensor computed as a function of the state of the $i-$th particle and the gradient at the integration point ij is computed as

$$\boldsymbol{D}_{ij} = \frac{1}{2}\left(\boldsymbol{D}_i + \boldsymbol{D}_j\right) \tag{3.14}$$

where the derivatives required to evaluate the viscous tensor are computed with MLS reconstruction. Note that since the MLS reconstruction is early performed for the convective terms, it does not require any additional reconstruction procedure to obtain a highly accurate discretization of the fluxes.

3.7. Validations cases

We present the numerical tests selected to assess the ability of the SPH-ALE-MOOD scheme to produce accurate and robust approximations. All the numerical examples have been computed using a third-order Runge-Kutta scheme for time integration.

3.7.1. 1D Riemann Problems

The first test cases are devoted to assess the stability and diffusive properties of the SPH-ALE-MOOD scheme. Here, we consider several one-dimensional tests.

The first test case is the 1D Riemann problem (R1) which is one of the four test cases proposed by Marongiu in [Marongiu, 2007]. In the context of SPH, the works presented in [Koukouvinis et al., 2013; Collé et al., 2019] also simulate this 1D Riemann problems with Tait EOS. The fluid is water modeled with Tait EOS ($\rho_0 = 1000\,kg/m^3$, $c_0 = 1466.0\,m/s$, $\gamma = 7$ and $p_0 = 0\,Pa$). The domain is $[0, 0.1]\,m$ and the initial condition is defined as

$$(R1) \qquad (\rho, u) = \begin{cases} (1100\,kg/m^3, 0\,m/s), & \text{if } x \leq 0.05\,m \\ (1000\,kg/m^3, 0\,m/s), & \text{otherwise} \end{cases}$$

A discretization of 100 particles is used and the solution is advanced up to time $t_{final} = 10^{-5}\,s$. The exact solution consists of a rarefaction wave traveling to the left and a shock wave traveling to the right. As a reference solution, we use the exact solution obtained with the algorithm given in [Toro, 2009] applied to the Tait EOS as indicated in [Ivings et al., 1998].

Figure 3.3 plots the pressure and velocity profiles obtained with the base scheme (first-order SPH-ALE scheme) [Vila, 1999] and with the SPH-ALE-MOOD model. The SPH-ALE base scheme smears the solution in the shock and rarefaction wave. As expected, for the same number of particles the SPH-ALE-MOOD provides a better representation of the shock front. The front and tail of the rarefaction wave provided for the

SPH-ALE-MOOD are also accurately captured and are free of overshoots near discontinuities. Both Eulerian and Lagrangian versions of the scheme produce very similar results, so we only plot here the results obtained with the Lagrangian description.

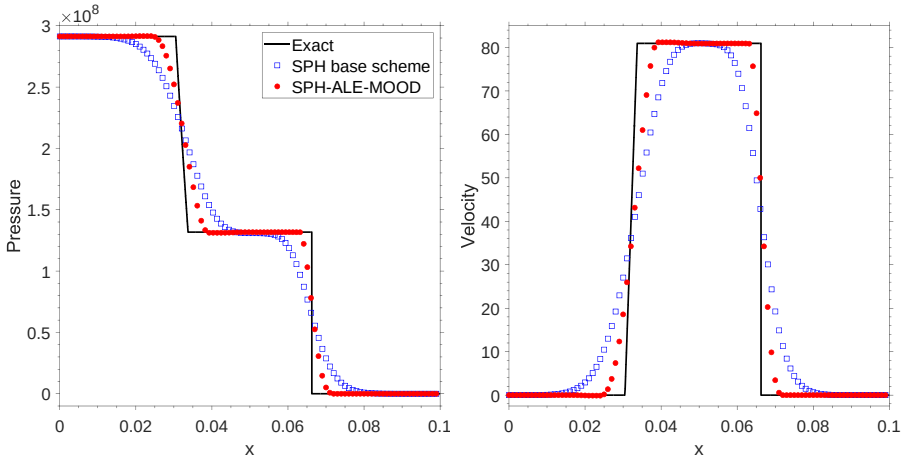


Figure 3.3. 1D shock tube problem (R1): Pressure and velocity at at time $t_{final} = 10^{-5}\,s$ using 100 particles in the domain $[0, 0.1]\,m$. Results obtained using the SPH base scheme (empty squares) and the SPH-ALE-MOOD method (filled circles).

We consider a second one-dimensional Riemann problem (R2). In this case, the liquid is assumed to follow the Tammann EOS ($\gamma = 7.15$ and $p_c = 3 \cdot 10^8\,Pa$). This test was proposed by Ivings and Toro in [Ivings et al., 1998] and has been also presented in [Pineda et al., 2019] with a SPH-ALE code with MUSCL reconstruction using a minmod limiter and a finer particle resolution.

The initial conditions for this problem are

$$(R2) \qquad (\rho, u, p) = \begin{cases} (1100\,kg/m^3, 500\,m/s, 5 \cdot 10^9\,Pa), & \text{if } x \leq 0.5\,m \\ (1000\,kg/m^3, 0, 1 \cdot 10^5\,Pa), & \text{otherwise} \end{cases}$$

The exact solution to this problem comprises a left-going rarefaction wave, a contact discontinuity and a right-going shock wave. The computational domain $[0, 1]$ is discretized with 200 particles.

In Figure 3.4 we plot the results for pressure, velocity, density, and internal energy at the final time $t_{final} = 7 \cdot 10^{-5}\,s$ obtained with the SPH base scheme and the SPH-ALE-MOOD using a Lagrangian description. The SPH-ALE-MOOD improves the results of the SPH base scheme in all the salient features present in the flow. In the density and internal energy plots, it is observed that the resolution of the contact discontinuity is not as sharp as the one obtained for the shock front. We note that the smearing in the contact discontinuity is inherent to the approximations made in the derivation of the Rusanov flux as reported in previous works [Avesani et al., 2014; Nogueira et al., 2016a].

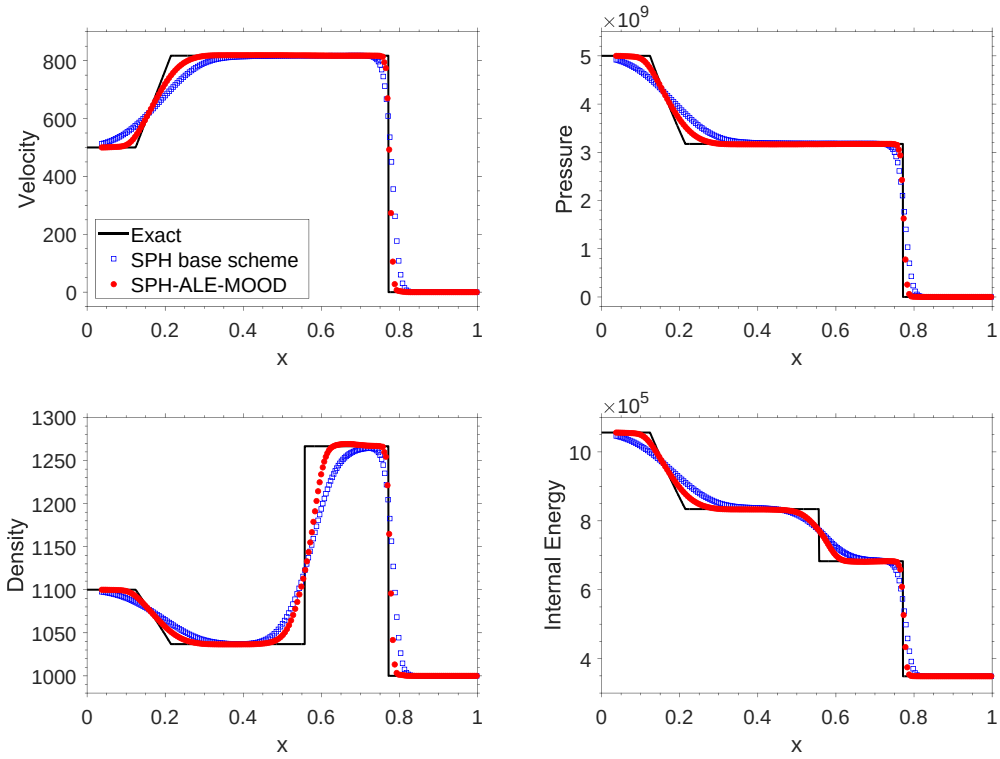


Figure 3.4. 1D Sod shock tube problem (T2): Simulations results at $t_{final} = 7 \cdot 10^{-5}\, s$ using 200 particles in the domain $[0, 1]\, m$. We plot velocity (top-left), pressure (top-right), density (bottom-left), and internal energy (bottom-right). Results obtained using the SPH base scheme (empty squares) and the SPH-ALE-MOOD (filled circles).

3.7.2. 2D Blast Explosion

The first two-dimensional test considered here is an extension of the one-dimensional shock tube R1 assuming cylindrical symmetry. The fluid is water modeled with Tait EOS($\rho_0 = 1000\, kg/m^3$, $c_0 = 1466.0\, m/s$, $\gamma = 7$ and $p_0 = 0\, Pa$). The computational domain is a circle of radius $R = 0.1\, m$ centered at the origin and the initial conditions are given by

$$(\rho, u, v) = \begin{cases} (1100\, kg/m^3, 0\, m/s, 0\, m/s), & \text{if } r \leq 0.05\, m \\ (1000\, kg/m^3, 0\, m/s, 0\, m/s), & \text{otherwise} \end{cases}$$

The configuration mimics an explosion with a shockwave traveling outwards and a rarefaction moving towards the origin. The reference solution is obtained by using a one dimensional finite volume code with a very fine mesh as explained in [Toro, 2009].

The evolution of the flow is simulated until $t_{final} = 10^{-5}\, s$ with the SPH-ALE-MOOD scheme.

We consider three different particle initializations to evaluate the effects of the initial positions of the particles on the quality of the numerical results. A radial distribution disposing particles in rings, a Delaunay distribution, which places particles in barycenters of triangles and finally, the third initial layout of particles is the result of applying a random displacement to the Delaunay distribution. The number of particles of the radial distribution (∼90,000) is slightly higher than the one of the Delaunay and Random distribution (∼75,000). Figure 3.5 shows the initial distributions considered.

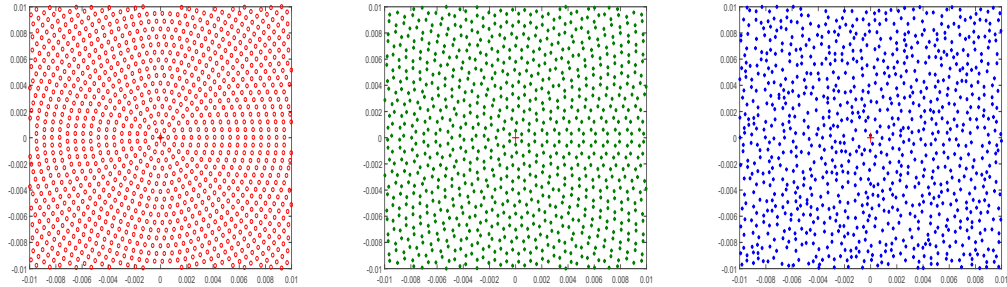


Figure 3.5. 2D Blast Explosion: Particle initializations. Radial distribution (left), Delaunay distribution (center), Random distribution (right).

Figure 3.6 shows the density at final time for the three initial particle layouts. The reference solution is represented with a black solid line. It is observed that the results preserve the radial symmetry of the physical problem.

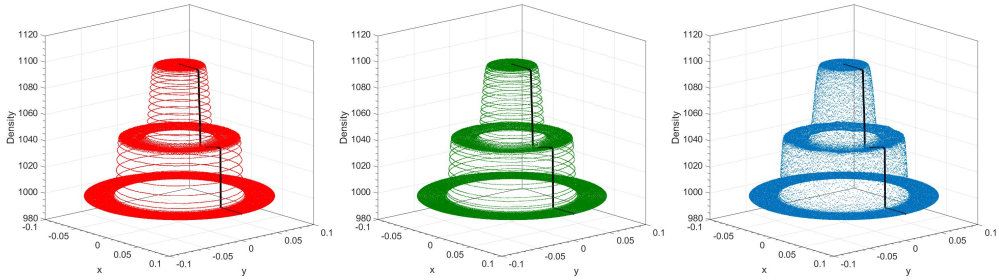


Figure 3.6. 2D Blast Explosion: Density plot in xy domain at time $t_{final} = 10^{-5}\,s$. Radial distribution (left), Delaunay distribution (center), Random distribution (right).

Figure 3.7 plots the density profiles along the radial coordinate. All the particles of the computational domain are represented and it can be noticed more clearly the ability of the model to preserve the radial symmetry. We note that the dispersion of the particles is really small for all particle distributions.

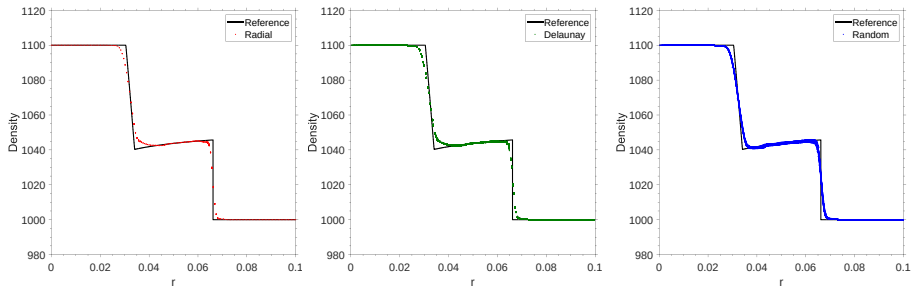


Figure 3.7. 2D Blast Explosion: Density profile along radial coordinate at time $t_{final} = 10^{-5}\,s$. Radial distribution (top left), Delaunay distribution (top right), Random distribution (bottom).

3.7.3. Taylor–Green Flow

The Taylor–Green flow is a classical test for numerical methods for the simulation of viscous flows. It provides an exact solution of the incompressible Navier-Stokes equations in a periodic domain. See [Kundu et al., 2015; Taylor & Green, 1937] for details. The flow involves the decay of four counter-rotating vortices within the periodic region of size $L \times L$ as shown in Figure 3.8.

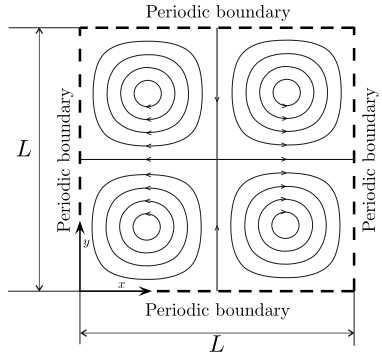


Figure 3.8. Taylor–Green vortex: Schematic representation of the computational domain and streamlines.

The exact solution is given in [Vittoz et al., 2019] and reads

$$\frac{u}{U_0} = \sin\left(\frac{2\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right)\exp\left(-\frac{8\pi^2}{Re}\frac{U_0}{L}t\right) \tag{3.15}$$

$$\frac{v}{U_0} = -\cos\left(\frac{2\pi x}{L}\right)\sin\left(\frac{2\pi y}{L}\right)\exp\left(-\frac{8\pi^2}{Re}\frac{U_0}{L}t\right) \tag{3.16}$$

$$\frac{p}{\frac{1}{2}\rho_0 U_0^2} = \frac{1}{2}\left[\cos\left(\frac{4\pi x}{L}\right) + \cos\left(\frac{4\pi y}{L}\right)\right]\exp\left(-\frac{16\pi^2}{Re}\frac{U_0}{L}t\right) \tag{3.17}$$

where Re is the Reynolds number of the flow, defined as $Re = \frac{\rho_0 U_0 L}{\mu_0}$. U_0 is a reference velocity magnitude, and ρ_0 and μ are constant values for the density and viscosity of the fluid, respectively.

A global decay kinetic energy factor, denoted by $r(t)$, is defined as the ratio of the overall kinetic energy at time t (E_k) and the corresponding one to initial time E_{k0}

$$r(t) = \frac{E_k}{E_{k0}} = \frac{\iint_D \frac{1}{2}[u^2(x,y,t) + v^2(x,y,t)]\,dx\,dy}{\iint_D \frac{1}{2}[u^2(x,y,0) + v^2(x,y,0)]\,dx\,dy} \tag{3.18}$$

Evaluation of Eq. (3.18) with the velocity field given by Eq. (3.15) and Eq. (3.16) results in an exponential decay according to

$$r(t) = \frac{E_k}{E_{k0}} = \exp\left(-\frac{16\pi^2}{Re}\frac{U_0}{L}t\right) \tag{3.19}$$

and by integration in the domain it can be derived that the initial kinetic energy is $E_{k0} = \frac{1}{4}U_0^2 L^2$.

For the simulations presented in this test, we consider a Taylor-Green flow with $L = 1\,m$,$U_0 = 1m/s$ and $\rho_0 = 1kg/m^3$. According to the weakly compressible approach, we assume that the fluid obeys the Tait equation with parameters $\rho_0 = 1\,kg/m^3$, $c_0 = 10\,m/s$, $\gamma = 7$ and $p_0 = 0\,Pa$. The case is simulated for $Re = 10$, $Re = 100$ and $Re = 1000$ with the Eulerian version of the SPH-ALE-MOOD scheme. Fluid particles are disposed inside the square domain on a Cartesian arrangement with $dx = dy = L/100$ as shown in Figure 3.9.

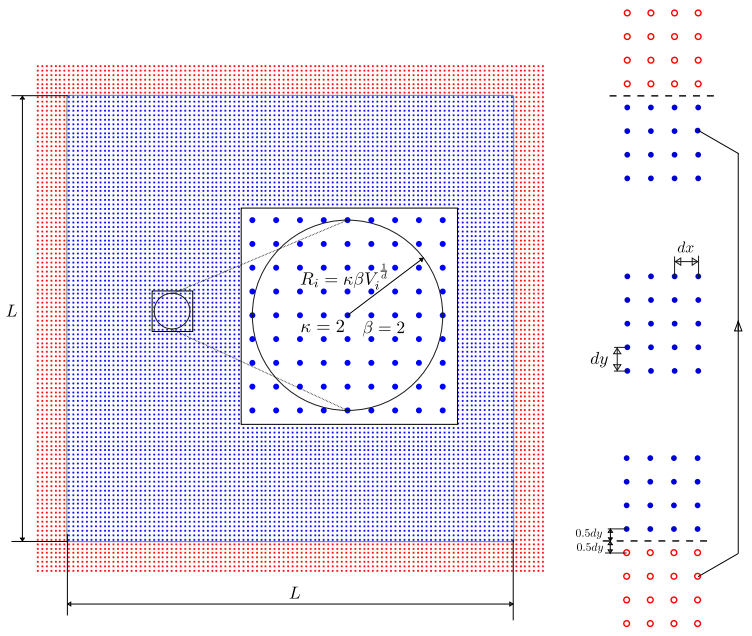


Figure 3.9. Taylor–Green flow: Layout of the particles. Hollow red circles: ghost periodic particles. Solid blue circles: fluid particles.

The initial conditions are computed using Eq. (3.15), Eq. (3.16) and Eq. (3.17) with the value of the density obtained from the Tait EOS for the analytical pressure.

Figure 3.10 shows the velocity components and pressure at non-dimensional time $t^* = tU_0/L = 1$. Velocity and pressure are smooth, similar to the analytical solution and no degradation of the vortical pattern is observed.

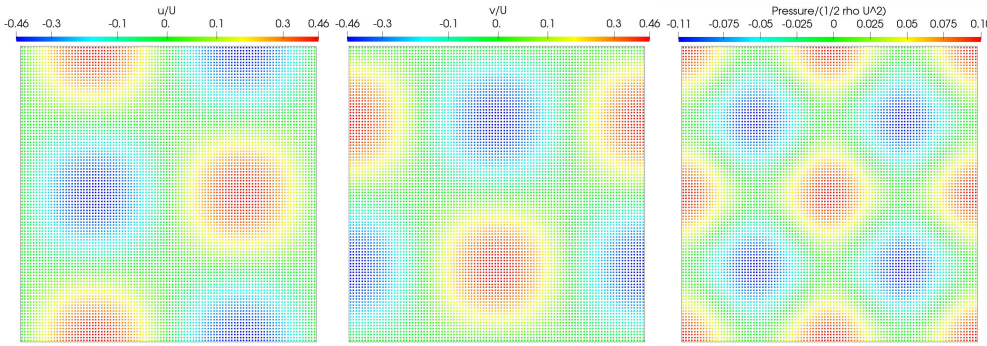


Figure 3.10. Taylor–Green flow: Results for $Re = 100$ at $t^* = 1$. Left: u-velocity field; center: v-velocity field: right: pressure field.

Figure 3.11 shows the time evolution of global decay of the kinetic energy, $r(t)$, and the maximum velocity modulus obtained using the SPH-ALE method and the corresponding reference incompressible solution for three different Reynolds numbers: $Re = 10$, $Re = 100$, and $Re = 1000$. The numerical results are in close agreement with the analytical solution, showing that the high-order reconstruction of the proposed scheme allows achieving a low dissipation scheme which is accurate for a wide range of Re numbers.

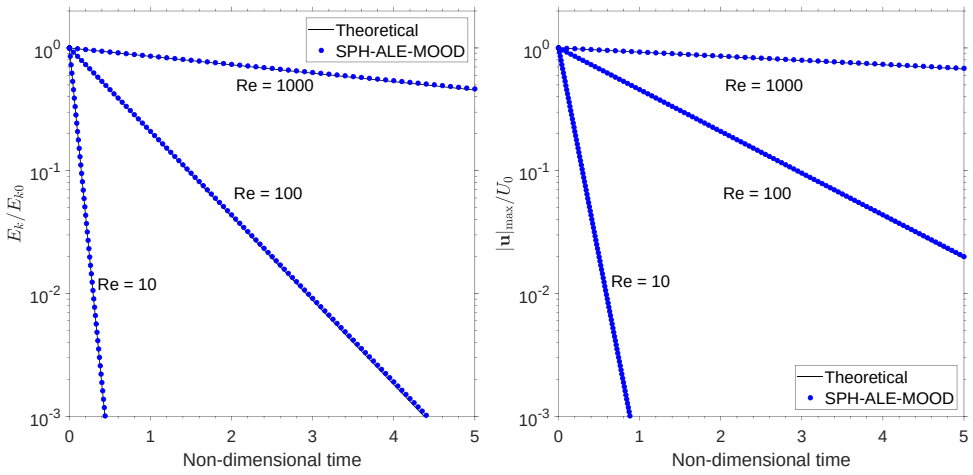


Figure 3.11. Taylor–Green flow: Comparison of the numerical and theoretical decay of the kinetic energy factor defined in Eq. (3.18) (left) and the maximum velocity (right) for $Re = 10$, $Re = 100$ and $Re = 1000$.

Following the work of Sun et al. [Sun et al., 2019], we have measured the time evolution of the pressure at the center of the domain for $Re = 100$ and $Re = 1000$ cases. The results are compared in Figure 3.12 with the theoretical solution and the solutions obtained with the δ-SPH and δ^+-SPH presented by Sun et al. [Sun et al., 2017, 2019]. We note that the proposed scheme shows better agreement with the reference solution for all particle resolutions. Moreover, it is remarkable the reduced amount of pressure oscillations, even for coarse discretizations.

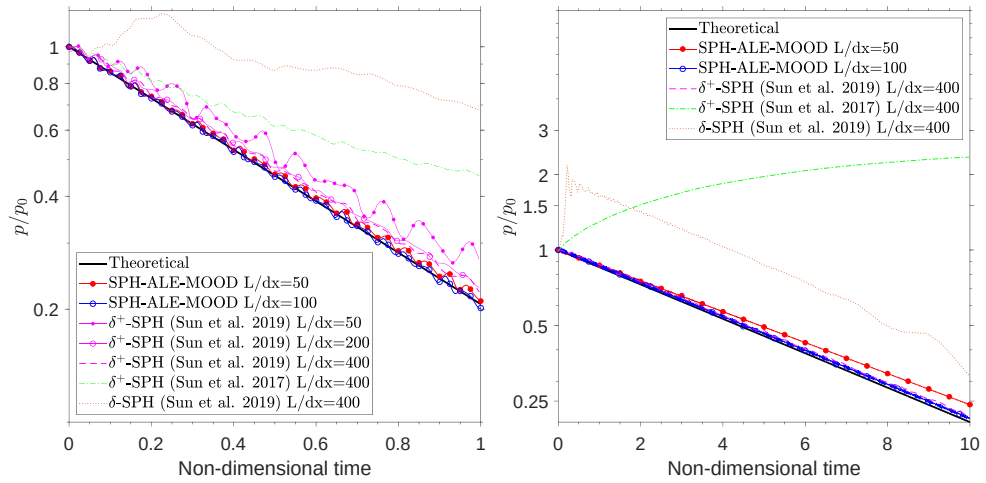


Figure 3.12. Taylor–Green flow: Comparison of the time evolution of the pressure at the center of the domain for $Re = 100$ (left) and $Re = 1000$ (right).

In Figure 3.13, the pressure field along $y = 0.5L$ is shown at $t^* = 6$ for $Re = 1000$. The results closely follow the theoretical solution even for the coarser discretization.

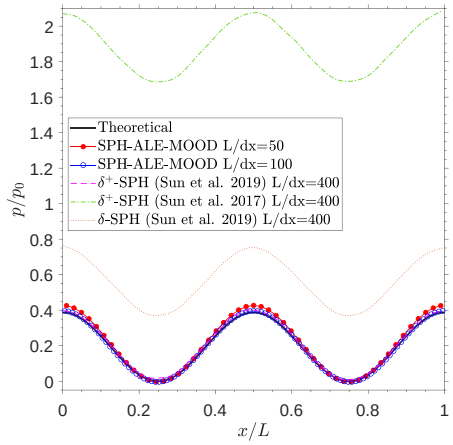


Figure 3.13. Taylor–Green flow: Comparison of the pressure field along $y = 0.5L$ for $Re = 1000$.

Concerning the computational cost of the proposed scheme, Figure 3.14 plots the CPU time consumed for different particle discretizations for a simulation until a final time of $t^* = 2$ for $Re = 100$. As expected, the Eulerian scheme is faster than the Lagrangian method. Then, a possible way for improving the efficiency of the proposed method is to combine Eulerian and Lagrangian particles. This idea has been explored previously in the context of ISPH [Fourtakas et al., 2018] and fits very naturally in the proposed formulation.

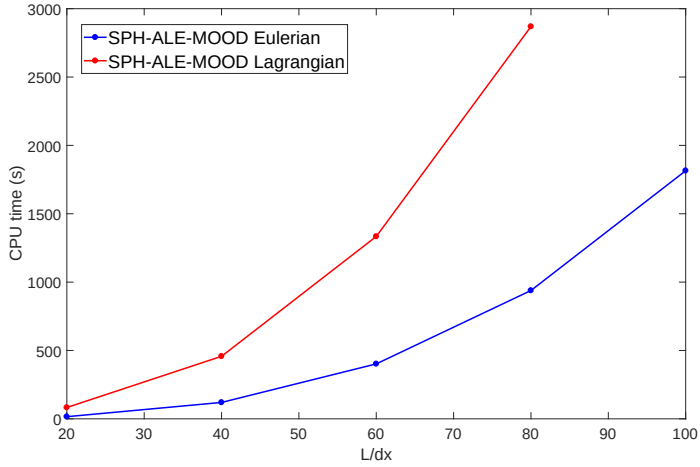


Figure 3.14. Taylor–Green flow:Computational costs for different particle discretization.

3.7.4. Couette and Poiseuille Flows

Couette and Poiseuille flows are special configurations of the incompressible Navier-Stokes equations that have analytical solution [Buresti, 2012]. In both cases, a Newtonian fluid moves between two infinite parallel plates. The Couette flow is driven by the movement of one of the plates whereas the Poiseuille flow is driven by a pressure gradient. As velocity does not vary along the flow direction, a finite length of the plates is considered with periodic boundary condition in left and right sides. Figure 3.15 shows the geometry model and boundary conditions considered for the simulations. In both configurations the fluid is initially at rest.

In the Couette flow, the time-dependent exact solution for the fluid velocity in the x-direction can be expressed as [Buresti, 2012; Morris et al., 1997]

$$u(y,t) = \frac{u_p}{L}y + \sum_{n=1}^{\infty} \frac{2u_p}{n\pi}(-1)^n \sin\left(\frac{n\pi}{L}y\right) \exp\left(-\nu\frac{n^2\pi^2}{L^2}t\right) \tag{3.20}$$

where u_p is the horizontal velocity of the upper plate and ν is the kinematic viscosity of the fluid.

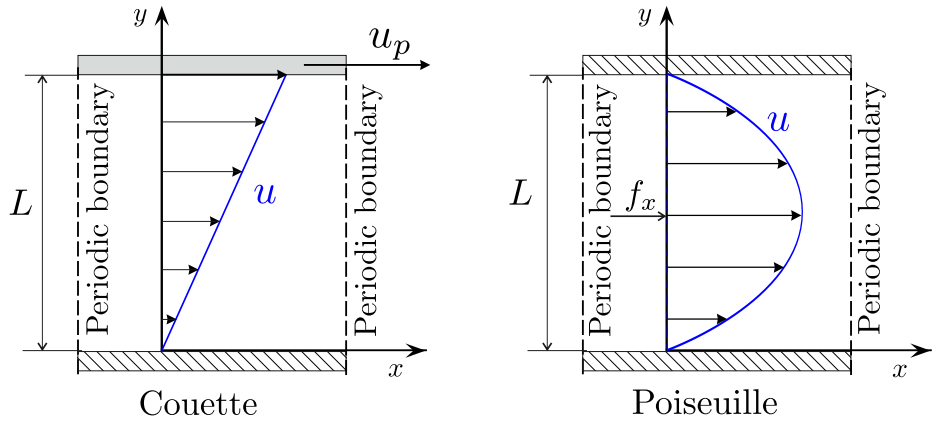


Figure 3.15. Couette and Poiseuille Flows: Schematic representation of the problems.

Similarly, for the Poiseuille flow the transient exact solution is given by [Buresti, 2012; Morris et al., 1997]

$$u(y,t) = \frac{f_x}{2\nu} y(y-L) + \sum_{n=0}^{\infty} \frac{4 f_x L^2}{\nu \pi^3 (2n+1)^3} \sin\left(\frac{\pi y}{L}(2n+1)\right) \exp\left(-\frac{(2n+1)^2 \pi^2 \nu}{L^2} t\right) \tag{3.21}$$

where f_x denotes a force source term in the x-momentum equation and, as such, it must be included in the source term $\boldsymbol{S}$ of the system of equations defined in Eq. (3.4). The force source term f_x and the steady peak velocity in the midplane of the channel u_{peak} are related by expression $u_{peak} = \frac{1}{8\nu} f_x L^2$.

For both problems, the Reynolds number is defined as $Re = \frac{u_{max} L}{\nu}$ considering the distance between plates L and the maximum velocity u_{max} as the reference length and velocity scales.

In this work, we conduct Couette and Poiseuille simulations for $Re = 10$. The same value was adopted in the works of Chiron [Chiron, 2017], Ferrand [Ferrand et al., 2013], and Fourtakas [Fourtakas et al., 2019]. The fluid is modeled using the Tait equation with $\rho_0 = 1\,kg/m^3$, $\gamma = 7$, $c_0 = 10\,m/s$, and $p_0 = 0\,Pa$. The kinematic viscosity considered for the fluid is $\nu = 0.1\,m^2/s$. The distance between plates is set to $L = 1\,m$ and half of this distance is considered for the periodic length in the flow direction.

In the Couette flow, u_p is set to $1\,m/s$, which leads to $Re = 10$. For the Poiseuille flow, the force source term is imposed as $f_x = 0.8\,m/s^2$ to produce the same Re number.

Figure 3.16 shows the arrangement of the particles employed for both the Couette and Poiseuille tests. The number of fluid particles between walls and periodic zones is 40 and 20, respectively, resulting in a squared arrangement with distance between particles $dx = L/40$.

In addition to the fluid particles, we need to incorporate ghost particles for implementing the periodic and wall boundary conditions. For the wall ghost particles, we follow the technique used in [Marrone et al., 2011]. A schematic representation of this technique is shown on the right of Figure 3.16. Dirichlet boundary conditions for velocity on the wall require that ghost particles update their velocity $\boldsymbol{u}^G = (u^G, v^G)^T$ following the vector equation.

$$\boldsymbol{u}^G = 2\boldsymbol{u}^W - \boldsymbol{u}^F \tag{3.22}$$

where $\boldsymbol{u}^F = (u^F, v^F)^T$ is the velocity of the mirroring fluid particle and $\boldsymbol{u}^W = (u^W, v^W)^T$ is the velocity vector of the wall. In case of fixed walls $\boldsymbol{u}^W = (0, 0)^T$ and for the top moving wall in Couette flow $\boldsymbol{u}^W = (u_p, 0)^T$.

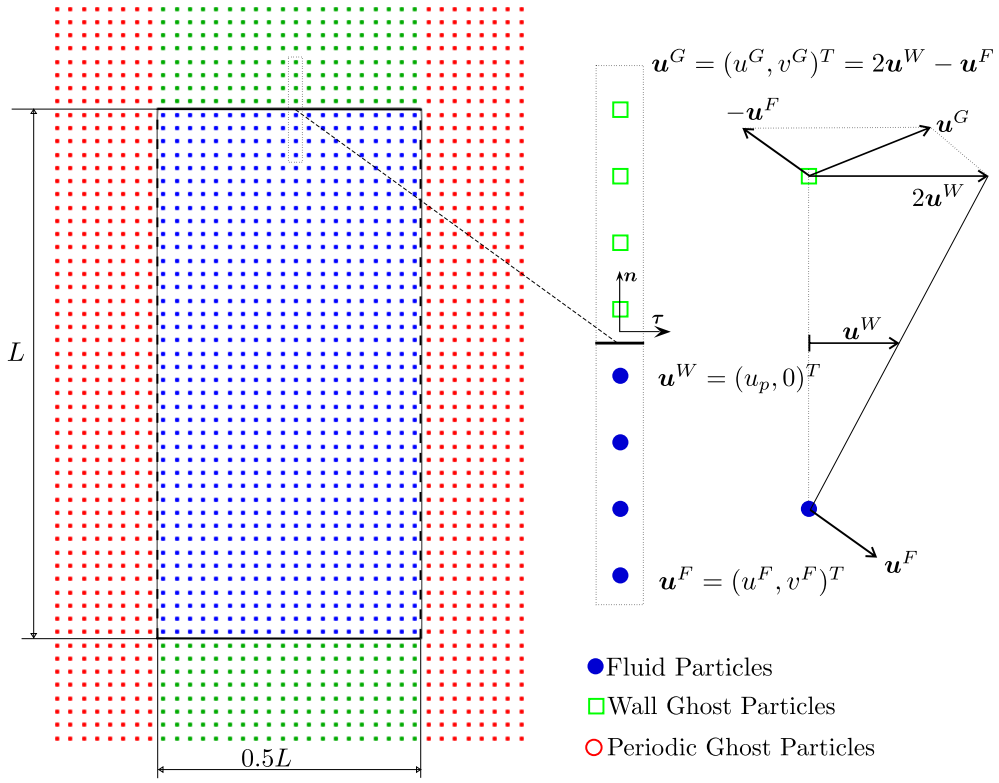


Figure 3.16. Couette and Poiseuille Flows: Particle layout (left). Schematic representation of the antisymmetric technique for wall ghost particles (right).

As we already have commented, one of the main advantages of the SPH-ALE method is the ability to use either Eulerian or Lagrangian description, and both configurations are able to obtain accurate solutions for this test case. Figure 3.17 shows the velocity profiles obtained with the SPH-ALE-MOOD scheme for Poiseuille flow at $Re = 10$ for the Eulerian and Lagrangian description. The exact solution is computed using

Eq. (3.21). For $t = 20\ s$ the flow is practically in the steady state condition and the obtained numerical solutions agree almost perfectly with the exact solution.

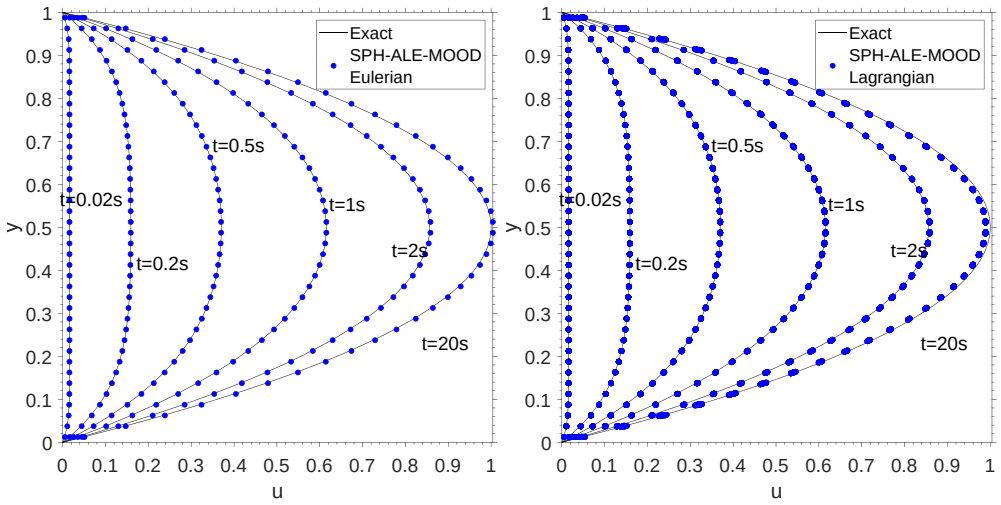


Figure 3.17. Time evolution of the velocity profile for the Poiseuille flow $Re = 10$ with Eulerian (left) and Lagrangian description (right). The numerical solution is compared with the exact solution presented in Eq. (3.21)**.**

Figure 3.18 shows the velocity profiles obtained with the SPH-ALE-MOOD scheme for the Couette flow at $Re = 1,\ 10,\ 100,\ 1000$. The exact solution is computed using Eq. (3.20). At $t = 0$, the velocity of the moving plate changes abruptly from rest to an horizontal velocity u_p forming a sharp discontinuity in the velocity field. A short time after that event, the obtained numerical results slightly deviates over the exact solution. This effect increases with the Reynolds number. For the last time instant, displayed for each Reynolds in Figure 3.18, the flow has practically reached the steady state, and the velocity profile is linear. The obtained results in the steady state are in close agreement with the exact solution for all the Reynolds numbers computed in this test case.

In Figure 3.18, the deviation from the reference solution observed in the first time instants of the simulations for $Re = 100$ and $Re = 1000$, are due to a lack of particles. For these Reynolds numbers, the spatial discretization is not able to capture the abrupt change in the velocity. To verify this, we plot in Figure 3.19 the results obtained for $Re = 100$ at $t = 0.2$ (left) and $Re = 1000$ at $t = 2$ (right) for different particle resolutions. It is seen that as the particle resolution increases the deviation is reduced, as expected.

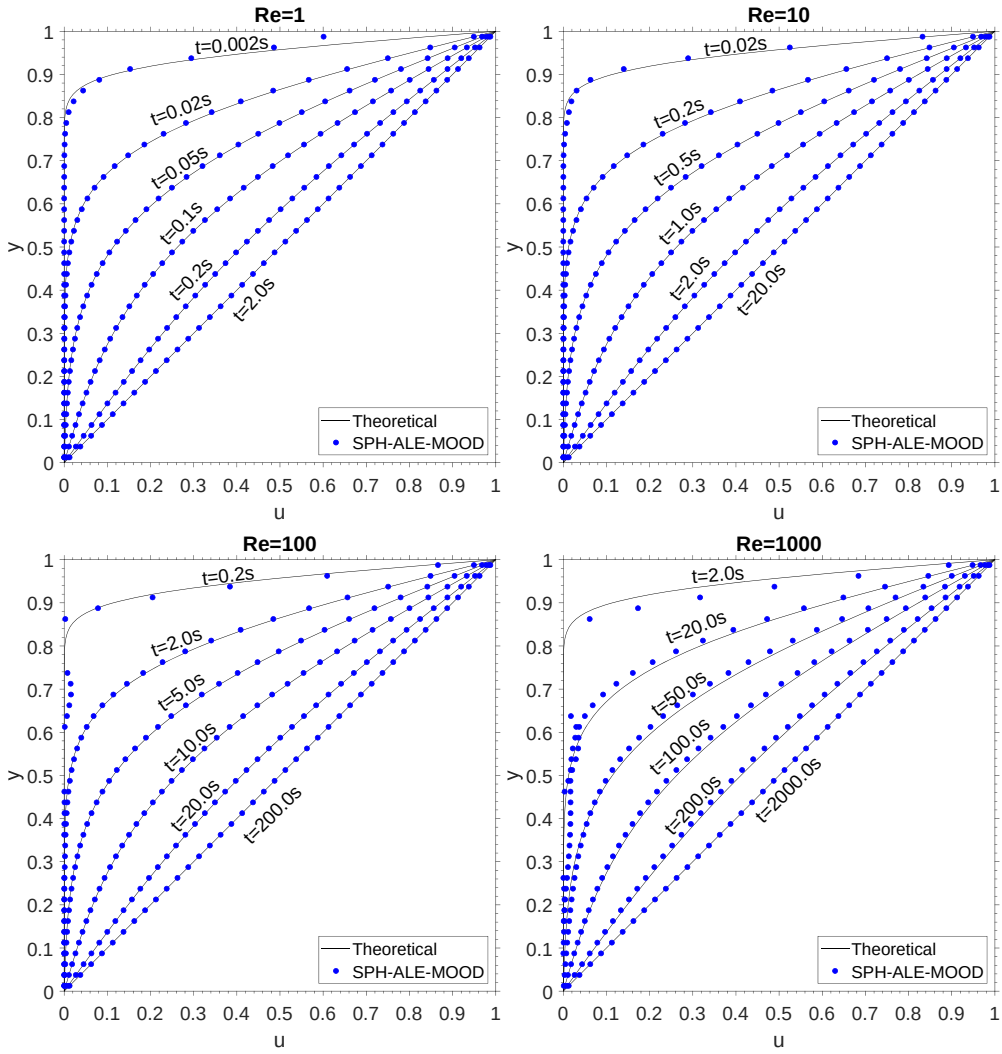


Figure 3.18. Time evolution of the velocity profile for the Couette flow $Re = 1$ (top left), $Re = 10$ (top right), $Re = 100$ (bottom left), and $Re = 1000$ (bottom right). The numerical solution is compared with the exact solution presented in Eq. (3.20).

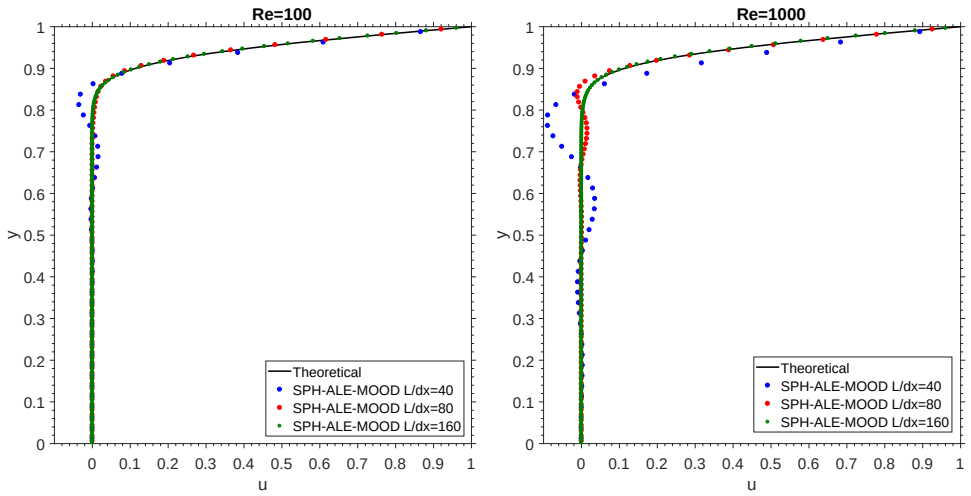


Figure 3.19. Convergence analysis for the Couette flow $Re = 100$ at $t = 0.2$ (left) and $Re = 1000$ at $t = 2$ (right).

3.7.5. 2D Lid-Driven Cavity Flow

The final test presented to assess the behavior of the proposed method is the 2D flow inside a square lid-driven cavity of length L. A schematic setup of the geometry is shown in Figure 3.20. The lateral and bottom walls are stationary, while the top wall moves horizontally to the right at speed u_w.

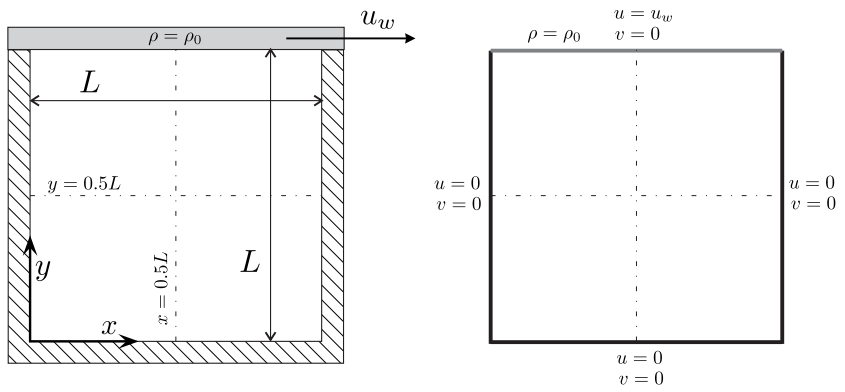


Figure 3.20. 2D Lid-driven cavity flow: Schematic representation of the geometry and boundary conditions.

Different Reynolds numbers, namely, $Re = 100$, $Re = 400$, and $Re = 1000$, are studied and results are compared to Ghia [Ghia et al., 1982]. As in the previous case, the velocity of the frame is set to zero adopting the Eulerian version of the SPH-ALE-MOOD scheme.

Figure 3.21 shows the layout and the type of particles. A Cartesian layout is adopted using a discretization of 100 particles on each side. Lid-driven cavity does not introduce any new type of boundary conditions, but the wall corners need to be taken into account to properly update ghost particle information, as schematically presented in Figure 3.21 on the top right corner.

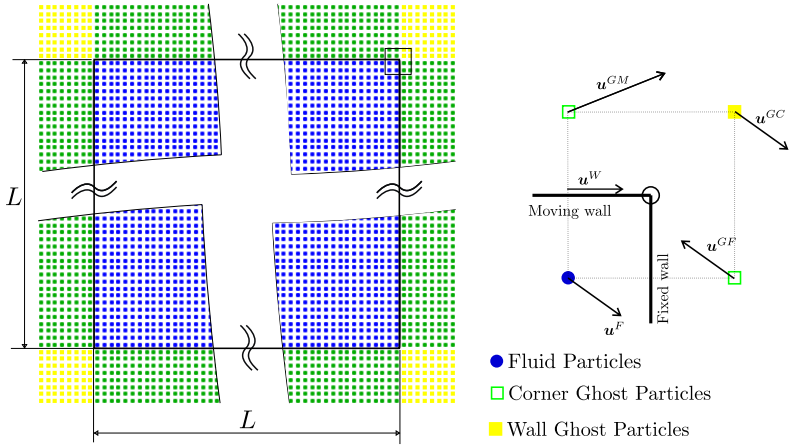


Figure 3.21. 2D Lid-driven cavity flow: Layout of the particles and treatment of ghost particles in corners.

In order to set the velocity for the corner particle $\boldsymbol{u}^{GC}$, we use a similar technique to the one proposed by Szewc et al. [Szewc et al., 2012]. Focusing on the top-right wall corner and considering the nearest four particles. We have one fluid particle with velocity $\boldsymbol{u}^{F}$, a ghost particle attached to the moving wall with velocity $\boldsymbol{u}^{GM}$, a ghost particle attached to the fixed wall with velocity $\boldsymbol{u}^{GF}$, and a ghost particle in the corner with velocity $\boldsymbol{u}^{GC}$. $\boldsymbol{u}^{F}$ evolves with the governing equations, and that $\boldsymbol{u}^{GM}$ and $\boldsymbol{u}^{GF}$ are updated according to Eq. (3.22). In order to set the velocity for the corner ghost particle $\boldsymbol{u}^{GC}$ we impose the velocity in the vertex of the corner as the average of the four particles.

Figure 3.22 shows the horizontal and vertical velocity profiles along the vertical and horizontal center line for $Re = 100$, $Re = 400$, and $Re = 1000$. Simulations were run for a $t_{final} = 500\,s$, clearly a time much longer than the one needed to reach the steady-state condition. Results are in good agreement with the reference solution [Ghia et al., 1982] for all the Reynolds number considered. Moreover, the obtained solutions are compared with the ones obtained by Lee et al. [Lee et al., 2008]. We note that the two schemes use the same number of particles for $Re = 400$. For $Re = 1000$, the ISPH scheme from [Lee et al., 2008] uses a finer discretization (160^2 particles).

The contours of the velocity magnitude superposed with the streamlines after $500\,s$ are shown in Figure 3.23. It is seen that the scheme is able to reproduce the primary and secondary vortices of the flow.

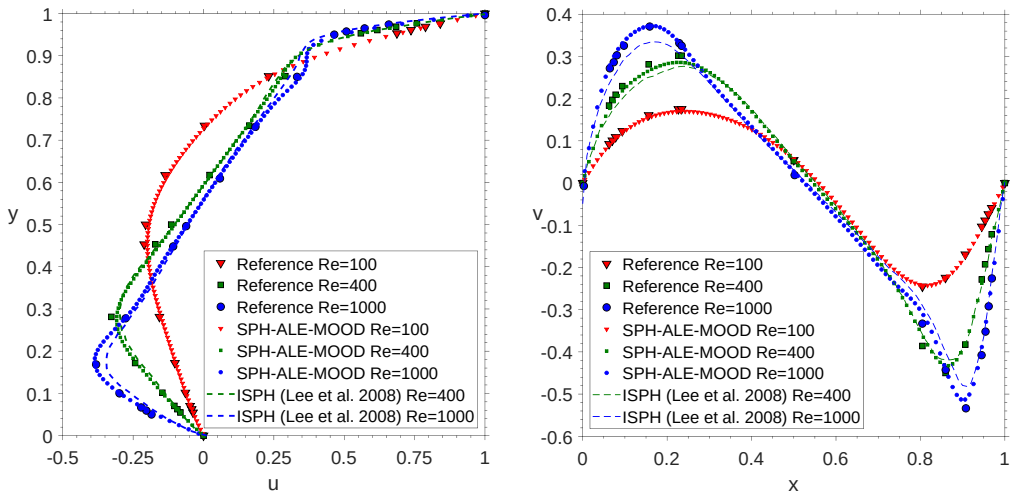


Figure 3.22. 2D Lid-driven cavity flow: On the left, horizontal velocity component u along $x = 0.5L$ for $Re = 100$, $Re = 400$, and $Re = 1000$. On the right, vertical velocity component v along $y = 0.5L$ for $Re = 100$, $Re = 400$, and $Re = 1000$.

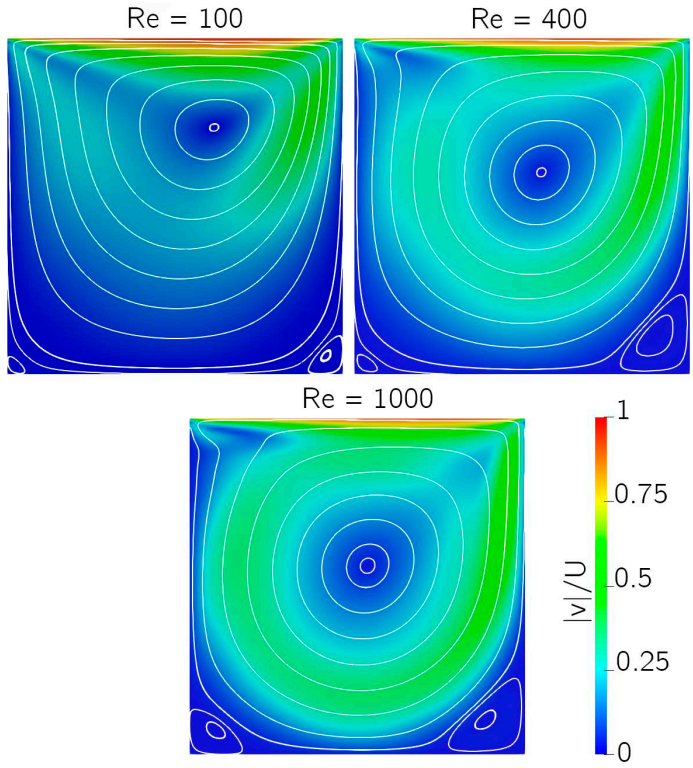


Figure 3.23. Contours of velocity and streamlines for lid driven cavity at different Re numbers. $Re = 100$ (top left), $Re = 400$ (top right), and $Re = 1000$ (bottom).

3.8. Conclusions

A high-accuracy SPH-ALE method for weakly compressible flow was implemented. The proposed method can deal with discontinuities. A new approach to compute the viscous flows terms is presented, where Moving Least Squares approximations are used to increase the accuracy and compute the derivatives needed for viscous fluxes. The performance of the proposed scheme is validated with a series of 1D and 2D benchmark problems, and compared with other WCSPH and ISPH schemes from the literature. The proposed method alleviates some of the known drawbacks of weakly compressible schemes. Thus, pressure oscillations are reduced compared with the $\delta-$SPH scheme, and there is no need for an special initialization of the particles. The proposed scheme obtains accurate solutions for any initial distribution of the particles. Moreover, the proposed formulation simplifies the coupling with grid-based methods (such as the FVM) or the combination of Eulerian and Lagrangian particles.

Chapter 4

The MLSPH-ALE method

4.1. Introduction

In this chapter a new meshless formulation is proposed to circumvent particle inconsistency issues of SPH-ALE. Instead of correcting the kernel particle approximation to restore zero and first order consistency we derived a meshless formulation that relies entirely on MLS approximation. The computational cost of creating MLS shape functions is similar to the cost of conducting a renormalization of the kernel gradient. Moreover, MLS approximation satisfies some desirable properties that can be conveniently exploited. The aim of this chapter is to develop a meshless method for the Navier-Stokes equations that keeps the advantages of the SPH-ALE method while circumventing particle consistency issues with MLS approximations, instead of modifying the kernel gradient. We named this new formulation as MLSPH-ALE method. A derivation of the model is provided starting from the Navier-Stokes equations given in conservative ALE form. Once the semidiscrete-system is derived, details are provided to recover the SPH-ALE formulation, showing the relationship between the two formulations. The introduction of MLS approximation brings also some additional advantages for the discretization of viscous fluxes and also for the reconstruction of the states to extend the MUSCL method [van Leer, 1979].

Meshless and mesh-based discretizations are usually considered as two competitive strategies to solve partial differential equations. In fact, if we consider the Finite Volume Method (FVM) and Smoothed Particle Hydrodynamics (SPH) as representative of these two approaches, both methodologies are far apart from each other. Vila [Vila, 1999] linked these methodologies by devising a weak version of a SPH method in an ALE framework where the interaction between particles is accounted by solving a Riemann problem. Junk [Junk, 2003] outlined that the FVM can be considered as a special case of meshless method satisfying the partition of unity property and posed the question if finite volume method really needs a topology for its implementation. Schaller et

al.[Schaller et al., 2013] analyzed three particle methods in decreasing similitude with the FVM and in terms of computational cost. The particle method with more similitude with the FVM is the one based on Voronoi tessellation, followed by the FVPM [Hietel et al., 2000] and the SPH-ALE proposed by Vila [Vila, 1999]. So, this two apparent different approaches actually have many features in common [Ivanova et al., 2013] suggesting that some successful techniques used in FVM can be incorporated in SPH and viceversa. In the last chapter of this dissertation, we will address this subject.

Despite the fact that meshless and mesh-based methods have advantages and drawbacks derived from the different form to accomplish the discretization of the partial differential equations, there are some physical phenomena (like turbulence) that are very computational demanding for both types of approaches. The research on the numerical simulation of turbulent flows has been mainly addressed by using mesh-based methods. Simulation of turbulence using meshfree methods has received less attention, and many studies deal with complex applications without providing details of their development, diffusion rates or details about the evolution of the energy spectrum. We refer the reader to [Monaghan, 2002, 2011; Mayrhofer et al., 2015; Hu & Adams, 2015; Di Mascio et al., 2017] for some examples.

In [Ramírez et al., 2018] the authors presented a new meshless approach (SPH-MLS) based on a Galerkin discretization of a set of conservation equations on an Arbitrary Lagrangian Eulerian approach, applied to the resolution of the Linearized Euler Equations. Instead of using kernel approximations, Moving Least Squares (MLS) [Lancaster & Salkauskas, 1981; Liu et al., 1997] were used as weight functions for the Galerkin discretization. The numerical method proposed was high-order and very accurate, and it was shown that it was able to be applied in the context of Computational Aeroacoustics. It can be seen as a general formulation which includes some well-known meshfree methods as a particular case, such as SPH-ALE [Vila, 1999; Avesani et al., 2014; Nogueira et al., 2016a] and Finite Point Method [Oñate et al., 1996b; Oñate, 1998; Ortega et al., 2009]. This formulation has several advantages over standard SPH methods. First, MLS functions form a partition of unity even in regions close to the boundaries. Moreover, shape function derivatives form a partition of nullity. These properties allow avoiding problems related with the initial position of particles which are found in other SPH methods [Colagrossi et al., 2012] and also lead to a higher accuracy than kernel usual approximations. Moreover, the number of neighbors required by the proposed formulation is smaller than that required when the kernel approximation is used [Ramírez et al., 2018], so the numerical scheme is more compact. In addition, the ALE behavior of the method proposed in this work makes possible the natural accommodation of Particle Shifting Techniques (PST). PST defines a drift velocity that moves the computational points to positions that produce more regular distributions and thus more accurate results [Nestor et al., 2009]. Dilts [Dilts, 1999, 2000] proposed different procedures to introduce MLS approximations to derive discrete meshless methods applied to a system of equations expressed in Lagrangian form. We note that the novelty of the present work is the use of the MLS approximation technique in a weak

form of the Navier-Stokes equations in ALE form.

In this chapter, we present several modifications to the method proposed in [Ramírez et al., 2018] and also the extension of this method to the resolution of compressible viscous flows. Thus, there are two main novelties in the formulation. First, we introduce a novel approach for the high-order reconstruction of the Riemann states. The usual Taylor polynomial reconstruction of the Riemann states is substituted by direct MLS approximations. This avoids the need of computing high-order derivatives for the Taylor reconstruction, but at the same time keeps the high-order accuracy of the reconstruction. The second novelty is related to the discretization of viscous terms. In the literature, there are different approaches to model the viscous part of the Navier-Stokes equations [Morris, 1996; Eirís et al., 2021]. The traditional approach in SPH methods is based on the definition of a number of discrete operators for approximating second derivatives. In this work we follow a different approach and approximate the viscous flux term with a formulation similar to that used for mesh-based methods in unstructured grids. Our approach is based on the use of MLS approximations of the Riemann states at both sides of the integration point. This approach fits perfectly in the SPH-MLS approach, since the MLS reconstruction is already computed.

This new discretization leads to a robust and very accurate meshless method, which is able to obtain results at least as accurate as grid-based methods of the same order of accuracy.

The structure of this chapter is as follows: firstly, we present the Navier-Stokes in conservative ALE form. Next, the proposed numerical method is introduced, including the novel discretization of viscous terms. Some numerical examples are presented to show the accuracy and robustness of the proposed formulation.

4.2. Governing equations

We define a transport operator $L_{\boldsymbol{w}}(\boldsymbol{U})$ with a regular velocity field $\boldsymbol{w}$ that operates on a vector of variables $\boldsymbol{U}$ giving

$$L_{\boldsymbol{w}}(\boldsymbol{U}) \equiv \frac{\partial}{\partial t}\boldsymbol{U} + \nabla \cdot (\boldsymbol{w} \otimes \boldsymbol{U}) \tag{4.1}$$

The conservative form of the Navier-Stokes equations can be compactly expressed using the transport operator $L_{\boldsymbol{w}}$ as

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \boldsymbol{\nabla} \cdot (\boldsymbol{F}_{\boldsymbol{w}} - \boldsymbol{D}) = \boldsymbol{S} \tag{4.2}$$

where the fluxes are split into a ALE hyperbolic-like part, $\boldsymbol{F}_{\boldsymbol{w}} = (\boldsymbol{F}^x_{\boldsymbol{w}}, \boldsymbol{F}^y_{\boldsymbol{w}}, \boldsymbol{F}^z_{\boldsymbol{w}})^T$, and an diffusive elliptic-like part, $\boldsymbol{D} = (\boldsymbol{D}^x, \boldsymbol{D}^y, \boldsymbol{D}^z)^T$. The right hand side contains the vector of source terms $\boldsymbol{S}$.

In 3D the vector of conservative variables reads as

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u^x \\ \rho u^y \\ \rho u^z \\ \rho E \end{pmatrix}, \tag{4.3}$$

where ρ is the density, $\boldsymbol{u} = (u^x, u^y, u^z)^T$ is the velocity vector, and E is the total energy.

The hyperbolic-like fluxes, also known as convective fluxes read in 3D as

$$\begin{gathered} \boldsymbol{F}^x_{\boldsymbol{w}} = \begin{pmatrix} \rho(u^x - w^x) \\ \rho u^x(u^x - w^x) + p \\ \rho u^y(u^x - w^x) \\ \rho u^z(u^x - w^x) \\ \rho E(u^x - w^x) + p u^x \end{pmatrix}, \; \boldsymbol{F}^y_{\boldsymbol{w}} = \begin{pmatrix} \rho(u^y - w^y) \\ \rho u^x(u^y - w^y) \\ \rho u^y(u^y - w^y) + p \\ \rho u^z(u^y - w^y) \\ \rho E(u^y - w^y) + p u^y \end{pmatrix}, \\ \boldsymbol{F}^z_{\boldsymbol{w}} = \begin{pmatrix} \rho(u^z - w^z) \\ \rho u^x(u^z - w^z) \\ \rho u^y(u^z - w^z) \\ \rho u^z(u^z - w^z) + p \\ \rho E(u^z - w^z) + p u^z \end{pmatrix} \end{gathered} \tag{4.4}$$

where p is the pressure. The elliptic or viscous fluxes read

$$\boldsymbol{D}^x = \begin{pmatrix} 0 \\ \tau^{xx} \\ \tau^{xy} \\ \tau^{xz} \\ \boldsymbol{\tau}^x \cdot \boldsymbol{u} - q^x \end{pmatrix}, \quad \boldsymbol{D}^y = \begin{pmatrix} 0 \\ \tau^{yx} \\ \tau^{yy} \\ \tau^{yz} \\ \boldsymbol{\tau}^y \cdot \boldsymbol{u} - q^y \end{pmatrix}, \quad \boldsymbol{D}^z = \begin{pmatrix} 0 \\ \tau^{zx} \\ \tau^{zy} \\ \tau^{zz} \\ \boldsymbol{\tau}^z \cdot \boldsymbol{u} - q^z \end{pmatrix}, \tag{4.5}$$

where $\boldsymbol{\tau}^x = (\tau^{xx}, \tau^{xy}, \tau^{xz})$, $\boldsymbol{\tau}^y = (\tau^{yx}, \tau^{yy}, \tau^{yz})$ and $\boldsymbol{\tau}^z = (\tau^{zx}, \tau^{zy}, \tau^{zz})$ represent the viscous stresses and $\boldsymbol{q} = (q^x, q^y, q^z)^T$ is the heat flux vector.

The viscous stress tensor is defined as

$$\boldsymbol{\tau} = \mu(\nabla(\boldsymbol{u}) + \nabla(\boldsymbol{u})^T) - \frac{2}{3}\mu\nabla \cdot (\boldsymbol{u})\boldsymbol{I} \tag{4.6}$$

where μ is the dynamic viscosity. These equations are completed with an equation of state (EOS), which in this chapter is the ideal gas EOS.

Once the governing equations have been presented we can remark some details. The hyperbolic flux tensor $\boldsymbol{F}_{\boldsymbol{w}}$ is not only dependent on the vector of conservative variables $\boldsymbol{U}$ but also on the transport velocity field $\boldsymbol{w}$, stated mathematically as $\boldsymbol{F}_{\boldsymbol{w}} = \boldsymbol{F}_{\boldsymbol{w}}(\boldsymbol{U}, \boldsymbol{w})$. The elliptic flux tensor $\boldsymbol{D}$ is a diffusive flux that does not depend on $\boldsymbol{w}$ velocity and it takes the form $\boldsymbol{D} = \boldsymbol{D}(\boldsymbol{U}, \nabla(\boldsymbol{u}))$. The transport velocity $\boldsymbol{w}$ is a field variable that is independent of the flow solution and can be prescribed at different

points of the domain according to different criteria. The transport velocity can be prescribed to move particles avoiding some pitfalls of pure Lagrangian methods by using some Particle Shifting Technique [Nestor et al., 2009]. Particles in the vicinity of a free surface can be moved in a Lagrangian fashion meanwhile particles in the vicinity of a wall can adopt an Eulerian framework to ease the implementation of boundary conditions.

4.3. Derivation of the MLSPH-ALE method

4.3.1. Partition of Unity

A set of shape functions verifies the Partition of Unity (PU) when the following property holds for any point $\boldsymbol{x}$ in the domain Ω

$$\sum_j N(\boldsymbol{x}-\boldsymbol{x}_j,h)=\sum_{j=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_j,h)=1 \tag{4.7}$$

where n_x denotes the number of neighbors in a compact support centered in $\boldsymbol{x}$ and extension related with length h. PU property provides a subdivision of the domain Ω into a set of subdomains Ω_i. Each subdomain Ω_i is associated with a particle i and a shape function $N(\boldsymbol{x}-\boldsymbol{x}_i,h)$ that is non-zero only in that subdomain.

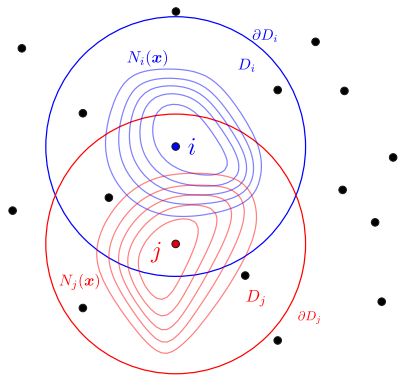


Figure 4.1. Isoline representation of particle shape functions N_i and N_j for two interacting particles i and j. Positions of particles i and j are represented by blue and red filled circles respectively. Other particles positions are represented by black filled circles.

Figure 4.1 provides a schematic representation of two particle shape functions $N_i = N(\boldsymbol{x}-\boldsymbol{x}_i,h)$ and $N_j = N(\boldsymbol{x}-\boldsymbol{x}_j,h)$. Some isoline values are represented to remark that shape functions do not have the symmetry property. Since the shape functions are usually associated with a kernel o weighted function we also depict the support domain D and its border ∂D.

The PU property assures the zeroth-order consistency meaning that any constant function c can be reproduced exactly

$$\hat{u}(\boldsymbol{x}) = \sum_j N_j(\boldsymbol{x})\, u_j = \sum_{j=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_j)\, c = c \sum_{j=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_j) = c \tag{4.8}$$

By taking the gradient operator at both sides of $\sum_{j=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_j) = 1$ and noting that ∇ is a linear operator it can be concluded that the derivatives of the shape functions verify also the Partition of Nullity

$$\begin{aligned} \nabla \sum_{j=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_j) &= \nabla 1 \\ \sum_{j=1}^{n_x} \nabla N(\boldsymbol{x}-\boldsymbol{x}_j) &= 0 \end{aligned} \tag{4.9}$$

It is possible to take advantage of the PU property when it is necessary to conduct integrals over the global domain Ω or over any subdomain Ω_i.

4.3.2. Volume integrals

PU property provides a subdivision of the domain Ω. The volume V of the global domain Ω can be expressed as the sum of the volume subdomains V_i as given by

$$V = \int_\Omega dx^d = \int_\Omega \sum_i N(\boldsymbol{x}-\boldsymbol{x}_i) dx^d = \sum_i \int_\Omega N_i(\boldsymbol{x}) dx^d = \sum_i V_i \tag{4.10}$$

The previous relation suggests to assign a volume weight V_i for i particle as the integral of the shape function $N(\boldsymbol{x}-\boldsymbol{x}_i)$ over the whole domain. Noting that the shape function takes non-zero values only in its compact support D_i

$$V_i = \int_\Omega N(\boldsymbol{x}-\boldsymbol{x}_i) dx^d = \int_{D_i} N(\boldsymbol{x}-\boldsymbol{x}_i) dx^d \tag{4.11}$$

It is important to remark that the volume weight of the particle V_i is smaller than the volume of the particle support D_i. Despite the overlapping between the support of the particles, the sum of the volume particle weights equals the volume of the global domain.

4.3.3. Volume integrals of functions

In the case of the integral of a general function $f(\boldsymbol{x})$ over the whole domain Ω, we use the MLS approximation given by $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n_x} N(\boldsymbol{x}-\boldsymbol{x}_i) f_i$ to obtain

$$\int_\Omega f(\boldsymbol{x}) dx^d \approx \int_\Omega \sum_i N(\boldsymbol{x}-\boldsymbol{x}_i) f_i dx^d = \sum_i f_i \int_\Omega N_i(\boldsymbol{x}) dx^d = \sum_i f_i V_i \tag{4.12}$$

In the following, we also assume that for any smooth function $f(\boldsymbol{x})$, we can apply a one-point quadrature approximation at the particle i

$$\int_\Omega f(\boldsymbol{x})\, N(\boldsymbol{x}-\boldsymbol{x}_i)\, dx^d \approx \int_\Omega f(\boldsymbol{x}_i)\, N(\boldsymbol{x}-\boldsymbol{x}_i)\, dx^d = f_i\, V_i \tag{4.13}$$

4.3.4. Derivation of the MLSPH-ALE method

In order to develop the formulation for the Navier-Stokes equations is convenient to group the ALE flux tensor $\boldsymbol{F_w}$ and the diffuse tensor $\boldsymbol{D}$ in total flux tensor given $\boldsymbol{F} \equiv \boldsymbol{F_w} - \boldsymbol{D}$. Then, Eq. (4.2) is rewritten as

$$L_{\boldsymbol{w}}(\boldsymbol{U}) + \boldsymbol{\nabla}\cdot\boldsymbol{F} = \boldsymbol{S} \tag{4.14}$$

We start the derivation process by multiplying Eq. (4.14) by the MLS shape function $N_i = N(\boldsymbol{x}-\boldsymbol{x}_i, h_i)$ centered at particle i. Then, we take the volumetric integral over the domain Ω

$$\int_\Omega N_i\, L_{\boldsymbol{w}}(\boldsymbol{U}) dx^d + \int_\Omega N_i\, \nabla\cdot\boldsymbol{F}\, dx^d = \int_\Omega N_i\, \boldsymbol{S}\, dx^d \tag{4.15}$$

The divergence of the flux can be approximated using MLS as

$$\int_\Omega N_i\, L_{\boldsymbol{w}}(\boldsymbol{U}) dx^d + \int_\Omega N_i \sum_{j=1}^{n} (\boldsymbol{F})_j \cdot \nabla N_j\, dx^d = \int_\Omega N_i\, \boldsymbol{S}\, dx^d \tag{4.16}$$

Using a one point quadrature approximation (see Eq. (4.13)) Eq. (4.16) reads

$$V_i\, L_{\boldsymbol{w}}(\boldsymbol{U}_i) + \int_\Omega N_i \sum_{j=1}^{n_i} (\boldsymbol{F})_j \cdot \nabla N_j\, dx^d = V_i\, \boldsymbol{S}_i \tag{4.17}$$

By using the Generalized RTT (see Eq. (2.10)) the term $V_i\, L_{\boldsymbol{w}}(\boldsymbol{U}_i)$ can be expressed as a time derivative following the particle giving

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \int_\Omega N_i \sum_{j=1}^{n_i} (\boldsymbol{F})_j \cdot \nabla N_j\, dx^d = V_i\, \boldsymbol{S}_i \tag{4.18}$$

Assuming that $(\boldsymbol{F})_j$ are point values

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} (\boldsymbol{F})_j \int_\Omega N_i \cdot \nabla N_j\, dx^d = V_i\, \boldsymbol{S}_i \tag{4.19}$$

Integrating by parts and applying the divergence theorem, we obtain

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} (\boldsymbol{F})_j \left(\int_{\partial\Omega} N_i\, N_j \cdot \boldsymbol{n}\, dx^{d-1} - \int_\Omega N_j \cdot \nabla N_i\, dx^d \right) = V_i\, \boldsymbol{S}_i \tag{4.20}$$

where $\boldsymbol{n}$ is the unitary surface normal and $\partial\Omega$ is the boundary of the domain Ω.

If we add Eq. (4.19) and Eq. (4.20) and divide by two, we obtain

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \frac{1}{2} (\boldsymbol{F})_j \left(\int_{\partial\Omega} N_i \, N_j \cdot \boldsymbol{n} \, dx^{d-1} - \int_{\Omega} N_j \cdot \nabla N_i \, dx^d + \int_{\Omega} N_i \cdot \nabla N_j \, dx^d \right) = V_i \, \boldsymbol{S}_i \quad (4.21)$$

Now, we add $\frac{1}{2}\boldsymbol{F}_i$ and subtract $\boldsymbol{F}_i$ to the fluxes in Eq. (4.21) yielding to

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2} (\boldsymbol{F}_j + \boldsymbol{F}_i) - \boldsymbol{F}_i \right] \left(\int_{\partial\Omega} N_i \, N_j \cdot \boldsymbol{n} \, dx^{d-1} - \int_{\Omega} N_j \cdot \nabla N_i \, dx^d + \int_{\Omega} N_i \cdot \nabla N_j \, dx^d \right) = V_i \, \boldsymbol{S}_i \quad (4.22)$$

Note that the operations involved in the transition from Eq. (4.21) to Eq. (4.22) have no effect at the continuum level, since, using Eq. (4.9) it can be proved that

$$\sum_{j=1}^{n_i} (\boldsymbol{F})_i \int_{\Omega} N_i \, \nabla N_j \, dx^d = (\boldsymbol{F})_i \int_{\Omega} N_i \sum_{j=1}^{n_i} \nabla N_j \, dx^d = \boldsymbol{0} \quad (4.23)$$

Using a one-point quadrature for the volume integrals we arrive to

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2} (\boldsymbol{F}_j + \boldsymbol{F}_i) - \boldsymbol{F}_i \right] \left(\int_{\partial\Omega} N_i \, N_j \cdot \boldsymbol{n} \, dx^{d-1} - V_j \, \nabla N_{ij} + V_i \, \nabla N_{ji} \right) = V_i \, \boldsymbol{S}_i \quad (4.24)$$

where $N_{ij} = N(\boldsymbol{x}_j - \boldsymbol{x}_i, h_i)$ and $N_{ji} = N(\boldsymbol{x}_i - \boldsymbol{x}_j, h_j)$.

The semidiscrete-system of Eq. (4.24) contains a boundary term that enable us to impose boundary conditions without requiring any external technique. The discretization of the boundary integral is detailed during the setup of the numerical problems solved in this chapter.

Note also that Eq. (4.22) and Eq. (4.24) were obtained for the discretization of the Navier-Stokes equations. For the particular case of the Euler equations the total flux tensor given by $\boldsymbol{F} \equiv \boldsymbol{F}_{\boldsymbol{w}} - \boldsymbol{D}$ is coincident to the ALE flux tensor $\boldsymbol{F}_{\boldsymbol{w}}$.

In order to establish some fast comparisons with standard SPH methodology let us assume there is no rigid boundary, so the boundary integral is equal to zero, and the discretization obtained by MLSPH-ALE for the Euler equations read as

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2} \Big((\boldsymbol{F}_{\boldsymbol{w}})_j + (\boldsymbol{F}_{\boldsymbol{w}})_i \Big) - (\boldsymbol{F}_{\boldsymbol{w}})_i \right] (-V_j \, \nabla N_{ij} + V_i \, \nabla N_{ji}) = V_i \, \boldsymbol{S}_i \quad (4.25)$$

Now, we define the numerical flux as $\boldsymbol{G}_{ij} \approx \frac{1}{2}\big((\boldsymbol{F}_{\boldsymbol{w}})_j + (\boldsymbol{F}_{\boldsymbol{w}})_i\big)$ (the difference in the two terms is the numerical dissipation). Therefore, the system of conservation laws can be discretized as

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\boldsymbol{G}_{ij} - (\boldsymbol{F}_{\boldsymbol{w}})_i\right] \left(-V_j \nabla N_{ij} + V_i \nabla N_{ji}\right) = V_i \boldsymbol{S}_i \tag{4.26}$$

If an Eulerian configuration is chosen for a set of particles distributed with equally weights V_i, Eq. (4.26) is very close to that obtained in [Bajko, 2013] using the Finite Point Method (FPM) for the discretization of the LEE. However, there are some differences, such as the symmetrization of the gradient of the shape functions, which is not performed in the FPM.

It is also interesting to remark that, if we use a kernel approximation instead of MLS shape functions we arrive to

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} = -\sum_{j=1}^{n_i} V_i V_j 2 \left[\boldsymbol{G}_{ij} - (\boldsymbol{F}_{\boldsymbol{w}})_i\right] \cdot \nabla W_{ij} \tag{4.27}$$

due to the anti-symmetry property of the kernel gradient i.e: $\nabla W_{ij} = -\nabla W_{ji}$. This is the SPH-ALE formulation presented in [Vila, 1999] in the form proposed in [Avesani et al., 2014]. We note that using the presented formulation, the origin and the necessity of the term $(\boldsymbol{F}_{\boldsymbol{w}})_i$ is clearly explained.

To complete the ALE formulation is necessary to provide an equation for the movement of the particles and an equation for evolving the weights of the particles.

The equation of the particles motion reads as

$$\frac{d\boldsymbol{x}_i}{dt} = \sum_{j=1}^{n_i} N_{ij} \boldsymbol{w}_j \tag{4.28}$$

As in the traditional SPH, the equation of evolution of the particles volumes follows a similar rationale. For a generic function $f(\boldsymbol{x}, t)$ the generalization of the Leibniz rule gives

$$\frac{d}{dt}\int_\Omega f \, dx^d = \int_\Omega \frac{\partial f}{\partial t} \, dx^d + \int_{\partial\Omega} f \boldsymbol{w} \cdot \boldsymbol{n} \, dx^{d-1} \tag{4.29}$$

If we consider the constant function, $f(\boldsymbol{x}, t) = 1$, a control volume Ω equal to V_i and the velocity at the boundary equal to $\boldsymbol{w}$ we obtain

$$\frac{d}{dt}\int_{V_i} dx^d = \int_{\partial V_i} \boldsymbol{w} \cdot \boldsymbol{n} \, dx^{d-1} = \int_{V_i} \nabla \cdot \boldsymbol{w} \, dx^d \tag{4.30}$$

Introducing the MLS approximation,

$$\frac{d}{dt}\int_{V_i} dx^d = \sum_{j=1}^{n_i} \int_{V_i} \boldsymbol{w}_j \nabla \cdot N_{ji} \, dx^d \tag{4.31}$$

Following the same procedure as in the system of conservation laws we obtain

$$\frac{d\,V_i}{dt} = \sum_{j=1}^{n_i} \left[\boldsymbol{w}_{ij} - \boldsymbol{w}_i\right] \left(-V_j\,\nabla N_{ij} + V_i\,\nabla N_{ji}\right) \tag{4.32}$$

It is important to remark that the numerical scheme is formally second-order, even though the reconstruction of Riemann states is higher-order [Hopkins, 2015].

4.4. Numerical Discretization

In this section the numerical discretization of the system given by Eq. (4.2) is presented. The first step in the discretization process is the reduction of the continuum domain into a cloud of points. Figure 4.2 shows a discretized geometry corresponding to a continuous domain Ω with boundary $\partial\Omega$.

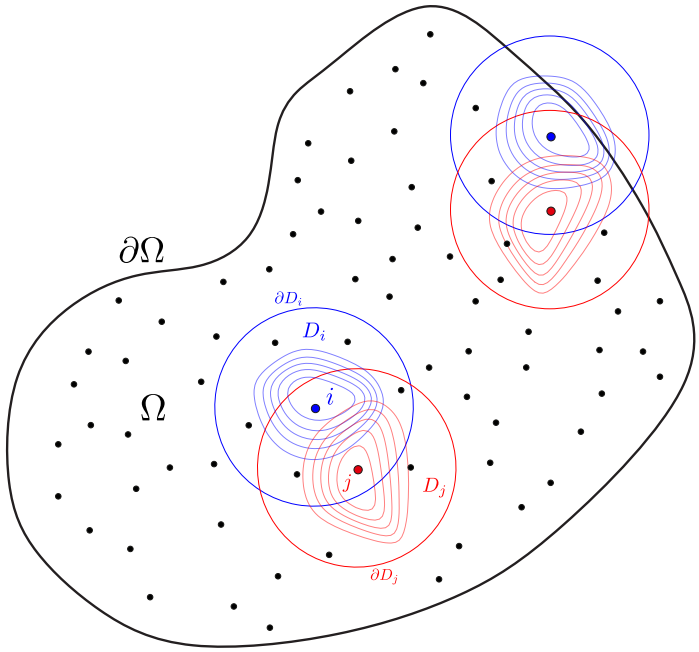


Figure 4.2. Computational domain Ω and kernel support D_i and D_j of particle i and j. Shape functions associated to particle i and j are represented by isocontour curves.

Each particle is labeled with an index i and we define its current position by means of its vector position $\boldsymbol{x}_i$ with respect to a Cartesian coordinate system. We denote by D_i the spherical support domain and by N_i the shape function associated to particle i. We use lowercase n_i for the number of particles inside the support domain of i particle. Note that each shape function N_i has a spherical compact support (a circle for 2D) but, in general, the shape function is not a centered radial function.

In Figure 4.2 two pairs of interacting particles are represented. One pair of interacting particles keeps their boundary support ∂D_i and ∂D_j inside the global domain Ω. The term $\int_{\partial\Omega} N_i\, N_j \cdot \boldsymbol{n}\, dx^{d-1}$ appearing in Eq. (4.24) is zero. On contrary, the boundary support of the other set of particles intersect the boundary of the global domain $\partial\Omega$ and the boundary term needs to be evaluated.

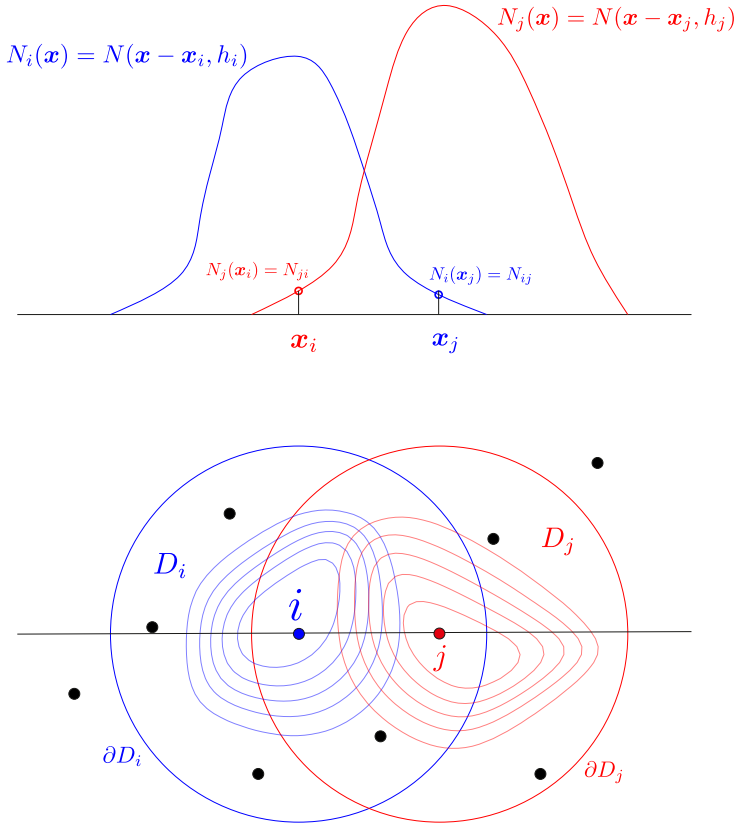


Figure 4.3. Shape functions associated to particle i and j are represented by isocontour curves.

Figure 4.3 depicts two neighboring particles in a two-dimensional domain. The bottom of the figure corresponds to a plan view meanwhile the top view shows the profile of the shape functions in a normal plane passing through the line that connects both particles. Note that we show the continuous version of the shape functions $N_i(\boldsymbol{x})$ and $N_j(\boldsymbol{x})$ although only the values in the locations of particles are required ($N_i(\boldsymbol{x}_j)$ and $N_j(\boldsymbol{x}_i)$).

Contrary to traditional SPH methods, where each particle is assigned with a constant mass value, in the proposed method the mass and volume for each particle are

variables that evolve with time.

The procedure to obtain the MLSPH-ALE semi-discrete system for the the Navier-Stokes equations was detailed in the previous section. Here, we start with the semi-discrete form valid for any interior particle i whose support does not intersect $\partial\Omega$ (see Figure 4.2). We recover Eq. (4.25), where the hyperbolic and elliptic fluxes were grouped in a total flux $\boldsymbol{F}$ exploiting the balance law form of the Navier-Stokes equations.

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2} \left(\boldsymbol{F}_j + \boldsymbol{F}_i \right) - \boldsymbol{F}_i \right] \left(-V_j \nabla N_{ij} + V_i \nabla N_{ji} \right) = V_i \boldsymbol{S}_i \tag{4.33}$$

The first term in the left side of the equation is the temporal derivative of the quantity carried by a general particle i. For a fixed particle with $\boldsymbol{w}_i = \boldsymbol{0}$ the temporal derivative term is equivalent to $\dfrac{\partial(V_i \boldsymbol{U}_i)}{\partial t}$. For a particle with $\boldsymbol{w}_i = \boldsymbol{u}_i$ the temporal derivative represents a material derivative $\dfrac{D(V_i \boldsymbol{U}_i)}{Dt}$. The second term is a sum over all neighbors of particle i and accounts for the hyperbolic and elliptic fluxes of particle i. We note that the usual kernel approximation has been substituted by a discretization based in MLS approximations.

In Eq. (4.33), we define $N_{ij} = N(\boldsymbol{x}_j - \boldsymbol{x}_i, h_i)$ and $N_{ji} = N(\boldsymbol{x}_i - \boldsymbol{x}_j, h_j)$. Note that these terms are required to obtain an anti-symmetric discretization of the effective area of the particle. By comparison with finite volume methods, the term $(-V_j \nabla N_{ij} + V_i \nabla N_{ji})$ plays the same role as the geometric interaction area between cells i and j.

The notation $N_i = N(\boldsymbol{x} - \boldsymbol{x}_i, h_i)$ stands for the MLS shape function evaluated at point $\boldsymbol{x}$ and centered at the position of particle i ($\boldsymbol{x}_i$). MLS shape functions also depend on a weighted function (see [Ramírez et al., 2017]), and the smoothing length h_i which is a measure of the compact support of the kernel function. The smoothing length associated to the particle i is variable for each particle and is computed as

$$h_i = \beta V_i^{\frac{1}{d}} \tag{4.34}$$

where d is the space dimensions number, V_i is the effective volume of the particle i and β is a constant parameter. The number of neighbors is variable and it depends on the value of β. In this work, we use $\beta = 1.5$ in the numerical applications. Note that this value differs from the one typically used in SPH-kernel approaches ($\beta = 2$). This implies that the number of particles involved in the approximation is reduced, which reduces the computational cost and also increases the accuracy of the method.

For the discretization of the Navier-Stokes equations we need to the decompose $\boldsymbol{F}$ flux into the ALE convective and the diffusive parts $\boldsymbol{F} = \boldsymbol{F}_{\boldsymbol{w}} - \boldsymbol{D}$. The ALE flux at the midpoint between two particles i and j is denoted as $(\boldsymbol{F}_{\boldsymbol{w}})_{ij} = \frac{1}{2}(\boldsymbol{F}_{\boldsymbol{w}})_j + \frac{1}{2}(\boldsymbol{F}_{\boldsymbol{w}})_i$. Similarly the diffusive flux at the midpoint is denoted as $\boldsymbol{D}_{ij} = \frac{1}{2}(\boldsymbol{D}_j + \boldsymbol{D}_i)$. Then, we can write

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[(\boldsymbol{F_w})_{ij} - (\boldsymbol{F_w})_i - (\boldsymbol{D}_{ij} - \boldsymbol{D}_i) \right] (-V_j \nabla N_{ij} + V_i \nabla N_{ji}) = V_i \boldsymbol{S}_i \quad (4.35)$$

Following [Ramírez et al., 2018] we introduce the numerical flux for the convective terms as an approximation of the flux at the midpoint between particles $\boldsymbol{G}_{ij} \approx (\boldsymbol{F_w})_{ij}$, and we obtain the final discretized form

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\boldsymbol{G}_{ij} - (\boldsymbol{F_w})_i - (\boldsymbol{D}_{ij} - \boldsymbol{D}_i) \right] (-V_j \nabla N_{ij} + V_i \nabla N_{ji}) = V_i \boldsymbol{S}_i \quad (4.36)$$

Moving Least Squares approximations are used also to provide high accuracy discretization of the convective and diffusive fluxes. We provide a brief exposition to introduce the nomenclature and remark some important points.

4.4.1. Convective Flux Discretization

In this work we have computed the convective flux, $\boldsymbol{G}_{ij}$, using the Rusanov numerical flux [Rusanov, 1962].

$$\boldsymbol{G}_{ij} = \frac{1}{2} \left[(\boldsymbol{F_w})^+_{ij} + (\boldsymbol{F_w})^-_{ij} \right] - \frac{1}{2} S^*_{ij} \Delta \boldsymbol{U}_{ij} \cdot \boldsymbol{n} \quad (4.37)$$

where S^*_{ij} is the maximum eigenvalue of the Jacobian matrix

$$S^*_{ij} = \max \left[(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{n} + c_{ij} \, , \, (\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{n} - c_{ij} \, , \, (\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{n} \right] \quad (4.38)$$

In Eq. (4.38), c_{ij} is the local speed of sound at the integration point. $(\boldsymbol{F_w})^-_{ij}$ and $(\boldsymbol{F_w})^+_{ij}$ denote the flux approximations of $\boldsymbol{F_w}$ on the left and right sides of the integration point (the two different Riemann states), and $\boldsymbol{n}_{ij}$ is the normal vector. The jump of the conservative vector is defined as $\Delta \boldsymbol{U}_{ij} = \hat{\boldsymbol{U}}^+_{ij} - \hat{\boldsymbol{U}}^-_{ij}$.

In order to achieve a high-order reconstruction of the Riemann states, the usual approach is to use a Taylor polynomial. Instead, we propose here the use of MLS approximations. Note that the MLS shape functions required for the reconstruction are already computed since they are used in the discretization of the governing equations, as indicated in Eq. (4.36). Thus, the cost of the evaluation of the reconstructed Riemann states is highly reduced, compared with the use of a Taylor polynomial (which requires an additional computation of high-order derivatives). Thus the reconstructed variables are computed as

$$\hat{\boldsymbol{U}}^-_{ij} = \sum_{k=1}^{n_i} N(\boldsymbol{x}_{ij} - \boldsymbol{x}_i, h_i) \boldsymbol{U}_k \quad \hat{\boldsymbol{U}}^+_{ij} = \sum_{k=1}^{n_j} N(\boldsymbol{x}_{ij} - \boldsymbol{x}_j, h_j) \boldsymbol{U}_k \quad (4.39)$$

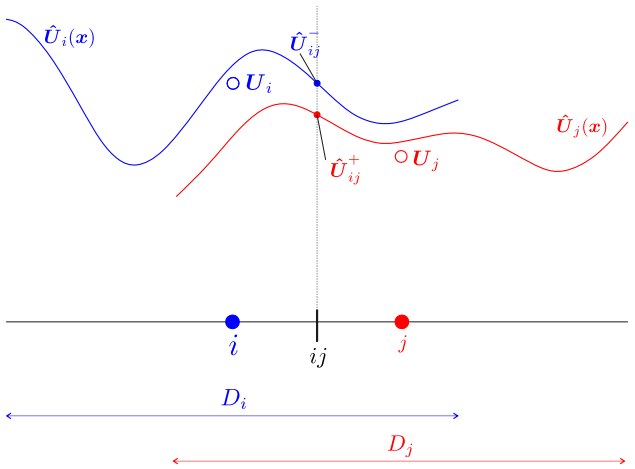


Figure 4.4. Schematic representation of the high-order Riemann states reconstruction at the integration point located in the midpoint between particles i and j.

4.4.2. Diffusive Flux Discretization

In this work, we use a novel formulation for the discretization of the diffusive fluxes, which is also based on the use of MLS approximations. MLS shape functions allows us to obtain high-accurate reconstructions of the variables on each stencil. Thus, it is possible to obtain two high-order approximations of the diffusive fluxes at the integration point using the two different states, and then compute the diffusive fluxes directly at each integration point (which is located at the midpoint between particles i and j), as the arithmetic mean of the diffusive fluxes of each Riemann state. A different approach is to center the MLS approximation on each integration point and perform a reconstruction for every integration as it was already proposed in the context of Finite Volume methods [Cueto-Felgueroso et al., 2007; Ramírez et al., 2014]. However, this approach will lead to higher computational cost due the large number of integration points and the need for computing the reconstructions on each time step. Thus, in this work, the diffusive fluxes are computed as

$$\boldsymbol{D}_{ij} = \frac{1}{2}(\boldsymbol{D}_{ij}^+ + \boldsymbol{D}_{ij}^-) \tag{4.40}$$

4.5. Numerical Results

This section presents the numerical results for several benchmark problems aimed at assessing the accuracy and efficiency of the proposed method for both steady and unsteady problems.

4.5.1. 2D viscous Taylor Green Vortex

The first test case is the 2D Taylor–Green Vortex. This test case is an exact time-dependent solution to the incompressible Navier-Stokes equations in 2D [Taylor & Green, 1937] and it is widely used to validate SPH methods [Hu & Adams, 2007; Sun et al., 2019]. The flow decays in time at a rate which is controlled by the viscosity. The analytical solution for the velocity and pressure fields corresponding to this flow in the domain $\Omega = [0, 2\pi] \times [0, 2\pi]$ is

$$\begin{aligned} u^x(x,y) &= e^{\frac{-2t}{Re}} \cos(y) \sin(x) \\ u^y(x,y) &= -e^{\frac{-2t}{Re}} \cos(x) \sin(y) \\ p(x,y) &= \tfrac{1}{4} e^{\frac{-4t}{Re}} (\cos(2x) + \cos(2y)) \end{aligned} \tag{4.41}$$

For Lagrangian particle methods this is a very challenging problem, since errors in the updating of the position of particles may lead the numerical scheme to reach a wrong solution. This fact will break the stationary closed trajectories which are obtained for the analytical solution. In this test case, the Reynolds number is set to 100. The computational domain Ω is discretized with 2500 particles disposed in a Cartesian distribution and periodic boundary conditions are imposed in all the boundaries. The initial configuration of particles and the initial velocity field are represented in Figure 4.5 a). In this problem, a Lagrangian configuration is adopted and particle shifting is used as described in [Nestor et al., 2009]. It is important to note that with the proposed methodology it is not required the use of procedures such as the particle packing method [Colagrossi et al., 2012] to obtain a suitable initial distribution of the particles, which is required in SPH methods based on kernel approximations [Colagrossi et al., 2012; Eirís et al., 2021].

In Figure 4.5, we plot the velocity magnitude at $t = 0$, $t = 1$ and $t = 2$. The symmetry of the solution is kept and the stationary closed trajectories are conserved.

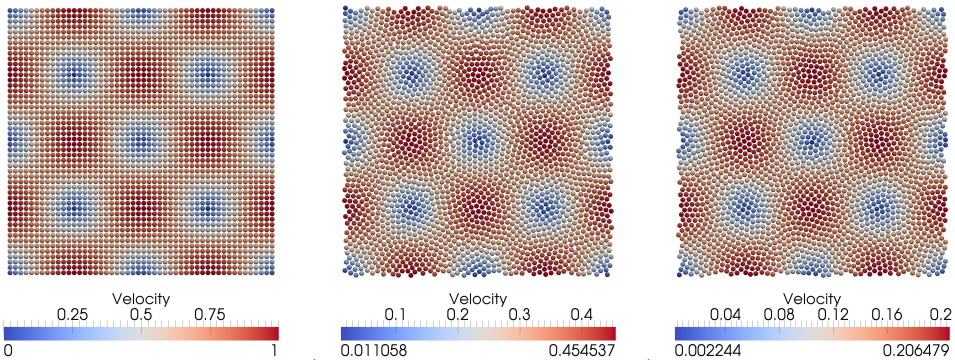


Figure 4.5. 2D Taylor–Green flow for $Re = 100$. Figures a), b) and c) represents velocity magnitude at $t = 0$, $t = 1$ and $t = 2$.

In Figure 4.6 the decay of the maximum velocity and the kinetic energy compared with the theoretical decay and with the results obtained with the δ^+-SPH method [Krimi et al., 2020; Sun et al., 2019] are presented. The results obtained by the proposed method show a perfect match with the theoretical curve. Note also that both, the Eulerian and the Lagrangian configurations, obtains similar results.

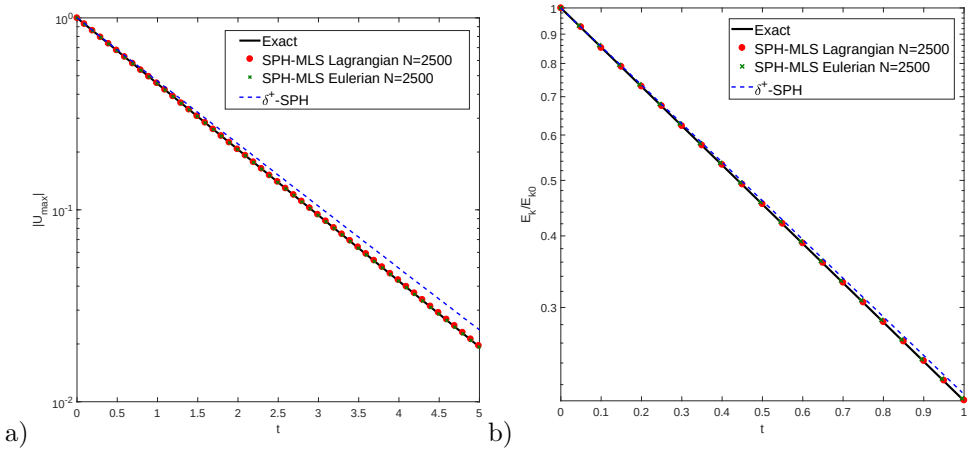


Figure 4.6. 2D Taylor Green flow for $Re = 100$. a) Time evolution of the maximum velocity magnitude over time compared with the exact solution and the δ^+-SPH results from [Krimi et al., 2020] and b) time evolution of the kinetic energy compared with the analytical solution and the δ^+-SPH results from [Sun et al., 2019]

4.5.2. Steady flow around a 2D cylinder $Re = 40$

In this test case, we consider the flow around a circular cylinder as a validation case of the proposed method for viscous flows with curved walls. The setup of this test case is as follows: The radius of the cylinder is $R = 0.5$, the Reynolds number is $Re = 40$ and the free-stream Mach number is $M_\infty = 0.1$.

The computational domain is discretized with a total number of 10800 particles and 120 particles around the cylinder. Figure 4.7 shows a close view of the particle distribution around the cylinder.

Free-stream variables are fixed at the outer boundary and we impose the no-slip and adiabatic boundary conditions at the cylinder solid wall. We solve the problem in an Eulerian configuration until the steady state is obtained. In Figure 4.8 the velocity field and the streamlines are plotted near the cylinder. It is observed that the solution obtained using the proposed numerical method is able to obtain good symmetry of the vortices created downstream the cylinder.

In Figure 4.9 the pressure coefficient distribution C_p around the cylinder is plotted at the steady state. The solution is compared with a reference solution obtained on a 3^{rd}-order finite volume scheme using a grid with the same spatial resolution than the

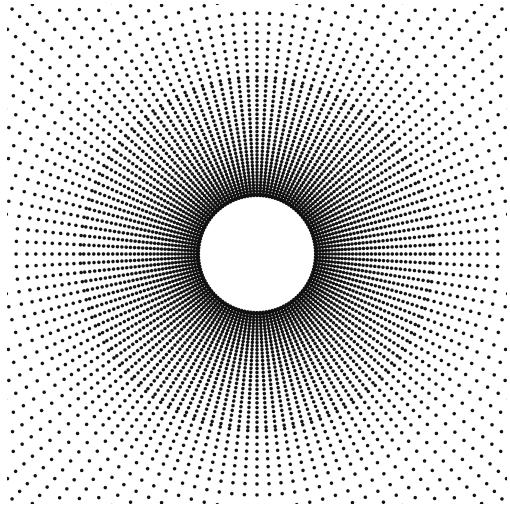


Figure 4.7. Steady $Re = 40$ flow around a 2D cylinder. Particle distribution around the cylinder.

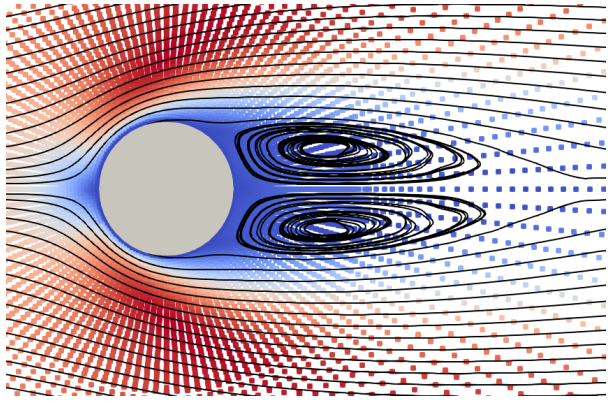


Figure 4.8. Steady $Re = 40$ flow around a 2D cylinder. Velocity field and streamlines around the cylinder.

present particle discretization [Ramírez et al., 2017]. It is observed that the proposed approach results are in excellent agreement with the reference solution.

The drag coefficient C_D and the front and back pressure coefficients ($C_p(0)$ and $C_p(\pi)$) are compared in Table 4.1 with reference values from the literature. It is observed that the proposed approach obtains results comparable to those obtained in the literature. In particular, they are in excellent agreement with the ones obtained with finite on a 3^{rd}-order finite volume scheme using a grid with the same spatial resolution [Ramírez et al., 2017].

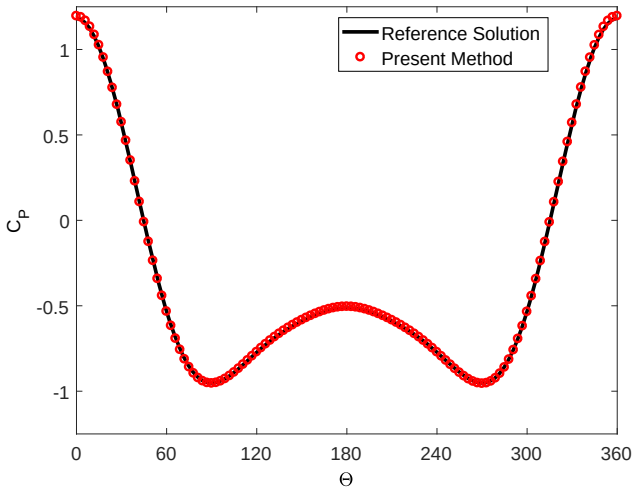


Figure 4.9. Steady $Re = 40$ flow around a 2D cylinder. Comparison of the pressure coefficient (C_p) distribution around the cylinder obtained using the proposed method and a 3^{rd}-order finite volume with the same spatial resolution [Ramírez et al., 2017].

Method	C_D	$C_p(0)$	$C_p(\pi)$
Present Method	1.570	−0.505	1.195
Ramirez et al. [Ramírez et al., 2017]	1.568	−0.512	1.180
Chassaing et al. [Chassaing et al., 2013]	1.565	−0.516	1.205
Niu et al. [Niu et al., 2003]	1.574	−0.555	1.147
He and Doolen [He & Doolen, 1997]	1.499	−0.487	1.133
Marrone et al. [Marrone et al., 2013]	1.6	–	–

Table 4.1. Steady $Re = 40$ flow around a 2D cylinder. Comparison of the drag coefficient C_D, front and back pressure coefficients ($C_p(0)$ and $C_p(\pi)$) obtained using the SPH-MLS scheme and other computations using grid methods [Ramírez et al., 2017; Chassaing et al., 2013], lattice Boltzmann methods [Niu et al., 2003; He & Doolen, 1997] and SPH methods [Marrone et al., 2013].

4.5.3. Unsteady flow around a 2D cylinder $Re = 100$ and $Re = 185$

In this test case, we address the simulation of a non-steady viscous flow past a two-dimensional cylinder. The domain and the particle distribution is the same than the used in the previous case. Now, two different Reynolds number are addressed, namely $Re = 100$ and $Re = 185$. This configuration leads to a flow with a periodic pattern of swirling vortices originated at cylinder wall.

Figures 4.10 and 4.11 show the temporal evolution of the drag and lift coefficients and the vorticity field respectively for the two different Reynolds numbers addressed. It is clearly observed the periodic pattern of vortices appearing downstream the cylinder.

Table 4.2 compares the drag and lift coefficients and the Strouhal number obtained using the proposed method with the results obtained by several authors. It is observed that the results obtained are comparable to those obtained using other methods.

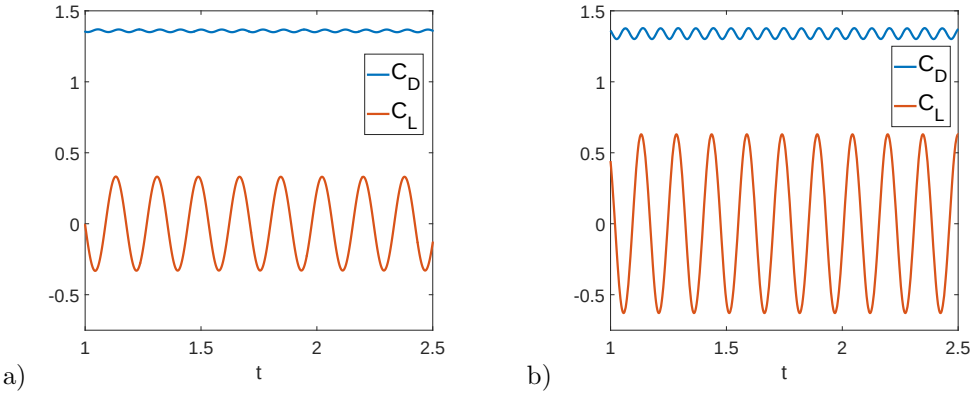


Figure 4.10. Unsteady flow around a 2D cylinder $Re = 100$ and $Re = 185$. Temporal evolution of the drag and lift coefficients, a) $Re = 100$ and b) $Re = 185$.

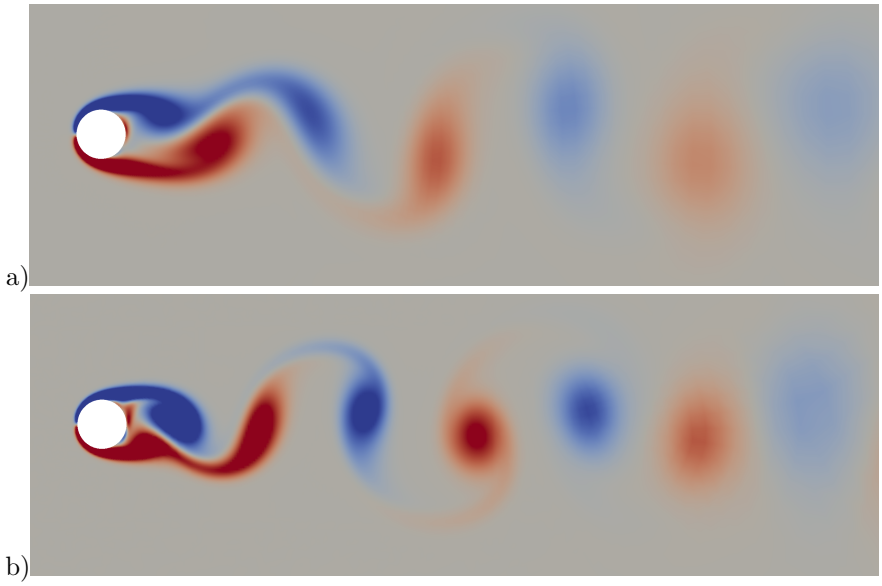


Figure 4.11. Unsteady flow around a 2D cylinder $Re = 100$ and $Re = 185$. Vorticity field, a) $Re = 100$ and b) $Re = 185$.

Conclusions

In this chapter we have proposed a high-accurate meshless discretization to solve the compressible Navier-Stokes equations. The standard kernel approximation is completely substituted by MLS approximations. In addition, the MLS approximations are also used to obtain a high-order reconstruction of the Riemann states instead of Taylor

	$Re = 100$			$Re = 185$		
Method	$\overline{C}_D$	C_L	S_t	$\overline{C}_D$	$C_{L,rms}$	S_t
Present Method	1.3587	±0.3325	0.1653	1.3398	0.4447	0.1937
[Liu et al., 1998]	1.350	±0.339	0.165	–	–	–
[Rajani et al., 2009]	1.3353	–	0.1569	–	–	–
[Marrone et al., 2013]	1.36	±0.24	0.168	–	–	–
[Ng et al., 2009]	1.368	±0.360	–	–	–	–
[Constant et al., 2017]	1.37	–	0.165	1.379	0.427	0.198
[Vanella & Balaras, 2009]	–	–	–	1.377	0.461	–
[Guilmineau & Queutey, 2002]	–	–	–	1.287	0.443	0.195
[Lu & Dalton, 1996]	–	–	–	1.31	0.422	0.195
[Liu et al., 2018]	–	–	–	1.372	0.427	–
[Sun et al., 2019]	–	–	–	1.363	–	0.196
[Liu & Hu, 2014]	–	–	–	1.289	0.451	0.197

Table 4.2. Unsteady flow around a 2D cylinder $Re = 100$ and $Re = 185$. Comparison of mean drag coefficient $\overline{C}_D$, mean and rms lift coefficients (C_L and $C_{L,rms}$) obtained using the SPH-MLS scheme and other computations using grid-based [Liu et al., 1998; Rajani et al., 2009; Guilmineau & Queutey, 2002; Lu & Dalton, 1996], immersed boundary [Ng et al., 2009; Constant et al., 2017; Vanella & Balaras, 2009; Liu & Hu, 2014], ALE [Sun et al., 2012] and SPH methods [Marrone et al., 2013; Liu et al., 2018].

polynomials. The proposed method presents several advantages compared with standard kernel approximations. Thus, the proposed formulation ensures the verification of the partition of unity even in regions close to the boundaries. Moreover, shape function derivatives form a partition of nullity, which avoids problems related with the initial configuration of particles. The use of MLS functions also leads to meshless schemes with greater accuracy than kernel approximations and with smaller stencils. We have shown that the proposed method is able to obtain comparable results to those obtained with grid-based methods in different regimes of viscous flow problems.

Chapter 5

MLSPH-ALE method and its relationship with some particle-based formulations

5.1. Introduction

In Chapter 4 MLSPH-ALE method was presented as a general method that contains other meshless formulations. The main purpose of this chapter is to place on the map the MLSPH-ALE method and to show its relationship with other widespread mesh-based and meshless methods. By adopting some particular settings, it is proved that MLSPH-ALE arrives at the same discretized equations obtained by other meshless methods. The study is restricted to the analysis of the Euler equations for a compressible fluid.

5.2. Types of numerical methods for Fluid Dynamics

As the first and the most widely used meshless method, there is some tendency in categorize as SPH to any particle based method that adopts a Lagrangian description. In the last decades, an extensive list of meshless formulations were published in the literature in order to circumvent some deficiencies or improve the performance of the original SPH formulation. Some of these new methods are Lagrangian particle-based methods using the kernel approximation, but there are others formulated in an Eulerian frame using other approximation techniques. Instead of organizing methods adopting the traditional SPH as a reference, we define three groups of numerical methods based on a more general concept. Following the ideas of Hopkins [Hopkins, 2015], an attractive distinction to organize numerical methods for simulating fluids consist in observing how the partition of the domain is accomplished.

In Figure 5.1, adapted from Hopkins [Hopkins, 2015], three different forms of subdividing the domain are represented by using a color palette code. For an irregularly set of particles, we focus the attention on three particles for which we have assigned red, green and blue color in their locations (black circles). The subdivision of the domain is represented visually by the color resulting from the addition of the primary colors RGB. Each color (Red,Green and Blue) represents the fraction of the volume at each point of the domain associated with each particle. In the first group of methods (on the left), the subdomain associated with each particle is represented by a plain-colored polygonal zone. This subdivision is associated with either a mesh or a Voronoi tessellation. In the third group of methods plotted (on the right), the subdivision of the domain is the result of each color decaying centered on each particle location. This is the type of geometric subdivision performed by traditional SPH methods based on the kernel approximation. The second group of methods is plotted in the center of the figure. In this group, the limits of each subdomain appear as blurry lines. The subdomains associated with each particle do not follow a centered pattern on particle location, as it occurs in methods of the third group. The subdivision of the domain corresponding with Group II is associated with meshless finite volume methods, that is, a class of meshless methods that share common features with FVM. Based on this hybrid nature, these methods will be referred hereafter as meshless-fv methods.

In each family we are ignoring the description of the flow (Lagrangian, Eulerian or ALE) and the formulation used to solve the equations (Compressible or Incompressible). For the reader interested in more complete classifications we refer to [Chen et al., 2017] and [Huerta et al., 2017].

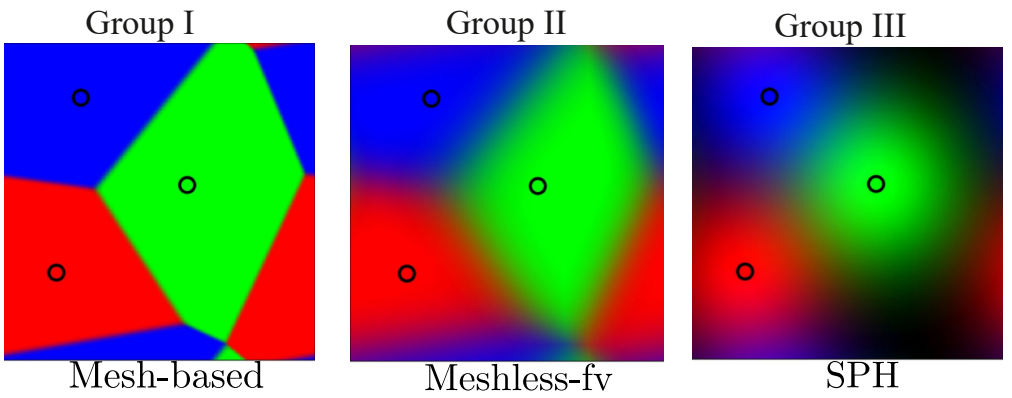


Figure 5.1. Groups of methods for Fluid Dynamics according to the partition of the domain. Left: Mesh-based methods with plain-colored polygonal subdomains. Middle: Meshless-fv methods with blurred polygonal subdomains. Right: SPH Methods with blurred subdomains centered at particle locations. Black circles denote the particle locations.

The classification provided considers an intermediate group that bridges the gap between mesh-based and SPH methods. For instance, particle disordering techniques used in Lagrangian SPH methods [Monaghan, 1989; Adami et al., 2013] have some

similarities with the use of the ALE framework to avoid degenerated cells in mesh-based methods. By using a meshless-fv method, Oger et al. [Oger et al., 2016] showed that particle disordering/shifting techniques need to be implemented within an ALE framework in order to avoid interpolation errors. Based on that findings, Antuono et al. [Antuono et al., 2021] incorporated the ALE framework into the δ-SPH, that was conceived originally as a pure Lagrangian SPH method. Another important transfer between different group of methods is related with the stabilization of the scheme. For instance, the stabilization terms in δ-SPH can be interpreted as diffusive terms of Riemann solvers in [Cercos-Pita et al., 2016]. As an example of the transfer of techniques from methods of third group to the first group we cite the scatter data approximation techniques that have been used in Finite Volume Methods to extend the order of reconstructions in high order Godunov methods [Ramírez et al., 2014].

Following we show the main features of the different groups. We remark that the MLSPH-ALE method developed in this thesis belongs to Group II and for that reason we will describe more succinctly methods in Group I and Group III.

5.2.1. Group I: Mesh-based methods

In the first group we include mesh-based methods like the Finite Volume Method, Finite Element Method and Particle Methods that use a Voronoi tessellation. In mesh-based methods the subdomain partition is the one that correspond with the mesh topology. In Voronoi methods the information is stored in a cloud of points. A tessellation algorithm is run to associate each particle with a subdomain whose geometry is clearly defined, and there is no overlapped zone between particle subdomains. In cell-centered Finite Volume methods the particle location is the barycenter of the cell meanwhile in the Voronoi tessellation the particle location is not coincident with the barycenter. In this group of methods each particle has a volume and area of the boundary that can be calculated with geometric formulae. Clear exposition of the FDM, FVM and FEM methods is given in Computational Fluid Dynamics textbooks [Hirsch, 2007], [Blazek, 2015], [Zienkiewicz et al., 2014]. For a further insight into Voronoi Tessellation Methods we recommend the publications associated with the AREPO project [Springel, 2010],[Muñoz et al., 2012].

5.2.2. Group II: Meshless-fv methods

In the second group of methods, we include particle methods in which the subdomain associated with each particle is calculated by means of functions that involve information contained in the neighborhood of the particle. Numerical methods in this group are meshless methods that share some characteristics with the FVM. In these methods the subdomain associated with each particle is not a geometrical entity. We can compute the volume associated with a particle but this is obtained by integration of a function which is not associated to the geometry of a cell. To remark this feature we use the term effective volume instead of the geometric volume used by mesh-based

methods. The same idea is used to define the interaction area associated with two neighboring particles. Most of the methods in this category are derived from the Euler equations expressed in conservative form. Among all the methods of this group, we can cite the following (non-exhaustive) list:

- Meshless Finite Volume Method (MFVM) [Ivanova et al., 2013]
- SPH-ALE [Vila, 1999]
- Renormalized SPH-ALE method [Lanson & Vila, 2008a,b]
- Finite Volume Particle Method (FVPM) [Hietel et al., 2000]
- Meshless Finite Volume (Gizmo-MFV) [Gaburov & Nitadori, 2011], [Hopkins, 2015][1]
- MLSPH-ALE [Ramírez et al., 2018]
- Moving Least Squares SPH (MLSPH) [Dilts, 1999, 2000]

The method proposed in this thesis lies in this group.

5.2.3. Group III: SPH methods

In the third group we include particle methods that approximate the continuum without requiring a geometric subdivision of the domain. The most representative method of this group is the SPH method. SPH is a Lagrangian method that assigns a fixed mass to each particle. As the numerical model updates the solution, particle density evolves and it is used to compute the volume associated to the particle. We note that this is a fictitious volume because SPH methods only require mass weights. In this group we include the original SPH method and some other formulations, like the δ-SPH and Riemann-SPH. Discretized equations for methods in this group are derived by replacing the continuous operators in the differential equations by discrete operators expressed in terms of the kernel function. We also include in this group particle methods like the Finite Point Method and the Finite Poinset Method. These methods do not use the kernel approximation and the cloud of points does not require mass weights. Among all the methods of this group, we can cite the following (non-exhaustive) list:

- Traditional SPH [Monaghan, 1988],[Monaghan, 2005]
- δ-SPH [Colagrossi & Landrini, 2003]
- Riemann-SPH [Inutsuka, 2002] ,[Parshikov & Medin, 2002]

[1]The MFV developed by Hopkins was implemented in the Gizmo project, see http://www.tapir.caltech.edu/ phopkins/Site/GIZMO.html. The Gizmo-MFV has also great similarities with the MFV proposed by Gaburov and Nitadori.

- Finite Point Method [Oñate et al., 1996b]
- Finite Pointset Method [Kuhnert, 2003], [Suchde et al., 2017]

5.2.4. Comparisons between methods

In the literature, comparisons between methods are usually done by conducting benchmarking studies over test cases [Few et al., 2016]. This chapter follows a different approach. We focus on the derivation process to highlight the existing connections among the different formulations. A similar strategy was adopted in the study conducted by Schaller et al. [Schaller et al., 2013].

The rest of the chapter is structured in the following manner. In a first step the attention is focused on showing the relations between the three groups. Then, in a second step, we show how the methods included in the second group can be derived by defining some particular settings in the MLSPH-ALE method.

5.3. Relations between groups of numerical methods

5.3.1. Mesh-based discretized equations

We derive the discretized equations of the FVM as a representative method of the mesh-based group. The conservative formulation of Euler equations with ALE description is written in integral form

$$\frac{d}{dt}\int_T \boldsymbol{U} dV + \oint_{\partial T} \mathbf{F}_{\boldsymbol{w}} \cdot d\boldsymbol{\Sigma} = \int_T \boldsymbol{S} dV \tag{5.1}$$

where T is the region of space associated with a cell and ∂T its boundary. $\boldsymbol{U}$ is the vector of conservative variables, $\mathbf{F}_{\boldsymbol{w}}$ is the ALE tensor flux and $\boldsymbol{S}$ the vector of source terms.

In Figure 5.2 we plot the centroids (filled points) of a group of cells. The cell boundary is given by a polygon when a bidimensional domain is considered. We note that the cell labeled with the index i has a finite number of neighboring cells n_i, that for the particular set depicted is $n_i = 6$. By focusing the attention on one of its neighbors, labeled with the index j, an enlarged view of cells T_i and T_j is shown on the right. We designate with $\boldsymbol{A}_{ij}$ the surface shared by neighboring cells i and j. We note that $\boldsymbol{A}_{ij}$ is a vector with a module equal to the area of the shared face and the direction given by the outward normal for cell i. It can be noticed that the direction of $\boldsymbol{A}_{ij}$ is in general not parallel with the line connecting the position of the centroids. As we are using the ALE flux tensor the expression is valid for fixed, moving and deforming cells. We could also use the same nomenclature for particle methods that employ a tessellation but in that case the full circles represent the generators and not the barycenter of the cells [Gaburro et al., 2020].

On an arbitrary cell T_i, Euler equation has a simple spatial discretization given by

$$\frac{d}{dt}\left(\bar{\boldsymbol{U}}_i V_i\right) + \sum_{j \in \partial T_i} (\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} = \bar{\boldsymbol{S}}_i V_i \tag{5.2}$$

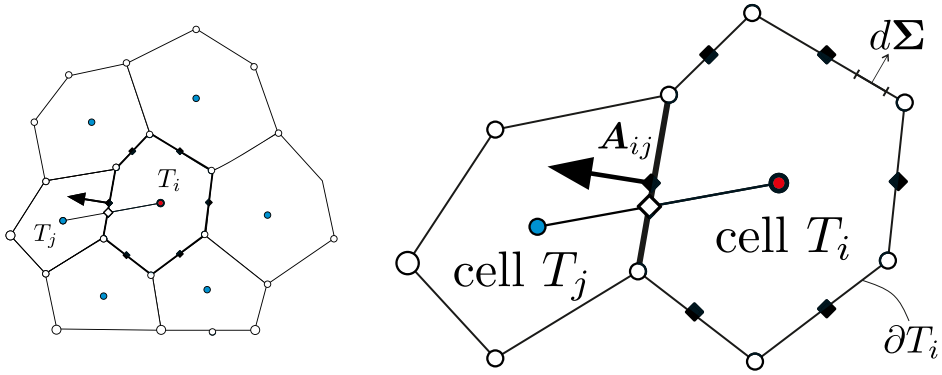


Figure 5.2. Discretization of the domain by the Finite Volume Method

Here $\bar{\boldsymbol{U}}_i$ and $\bar{\boldsymbol{S}}_i$ are the mean values of $\boldsymbol{U}$ and $\boldsymbol{S}$ inside a mesh cell T_i which has volume V_i [2]. Note tat the sum is carried over the boundaries of the cell T_i designated as ∂T_i. For convenience we introduce a compact notation for the surface integral of a single face separating cell i and j

$$(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} = \int_{\partial T_{i,j}} \mathbf{F}_{\boldsymbol{w}} \cdot d\boldsymbol{\Sigma} \tag{5.3}$$

This simple derivation used by FVM can be accomplished because the volume and boundaries of cell are geometric entities associated with the mesh. Unfortunately this fast procedure can not be followed in meshless methods, since in those methods particles represent a region of the continuous domain, but that region lacks any boundary definition to accomplish the integrals as expressed in Eq. (5.1). Thus, meshless methods must follow a different procedure to discretize the governing equations. Two of these procedures will be outlined in the following sections.

5.3.2. Meshless-fv discretized equations

As a representative example for meshles-fv methods we select the method proposed by Ivanova et al. [Ivanova et al., 2013]. In their work, an appendix is provided with title "On a spatially-discrete formalism for mesh-less finite-volume-methods". We use the acronym MFVM (Meshless Finite Volume Method) to refer to the method proposed in the appendix of their publication. The paper reviews hydrodynamic codes for cosmological applications where Lagrangian SPH and Cartesian Eulerian methods

[2] For 2D domains consider that $V_i \equiv \boldsymbol{A}_i \cdot \mathbf{1}$

were conceived as two totally different approaches. Authors proved that Lagrangian particle-based methods are a subset of more general meshless-fv schemes. The publication encourages to take a flexible approach in the development of numerical methods and opens the path for developing hybrid codes that exploit the advantages of both types of methods. A sentence given in the introduction is very elucidating stating that "Lagrangian methods can be implemented on unstructured Voronoi meshes and Eulerian schemes can be successfully formulated in entirely meshless form".

In [Ivanova et al., 2013], authors presented a generic formalism that leads to spatially discrete meshless finite volume equations, although the formalism is not implemented in a code and therefore no results are provided in the article. Procedure begins with the continuous weak form of Euler equations in the form given by Eq. (2.74)

$$\forall \varphi \in C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right) \qquad \int_{\mathbb{R}^d \times \mathbb{R}^+} \left(\boldsymbol{U} L^*_{\boldsymbol{w}}(\varphi) + \boldsymbol{F}^{\alpha}_{\boldsymbol{w}} \frac{\partial \varphi}{\partial x^{\alpha}} + \boldsymbol{S}\varphi \right) dx^d \, dt = \mathbf{0} \tag{5.4}$$

where the integral is taken over all the space-time domain. In order to ease the notation, the adjoint transport operator $L^*_{\boldsymbol{w}}(\varphi)$ is recognized as the advective derivative in the particle velocity field $\boldsymbol{w}$ and denoted more compactly as $\dot{\varphi} = \partial\varphi/\partial t + \boldsymbol{w} \cdot \nabla\varphi$. We also change from index notation to vector notation to write

$$\forall \varphi \in C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right) \qquad \int_{\mathbb{R}^d \times \mathbb{R}^+} \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi \right) dx^d \, dt = \mathbf{0} \tag{5.5}$$

To get a particle approximation of this ALE weak problem, a set of moving particles given by paths $\boldsymbol{x}_i(t)$ is considered (with index i varying from $i = 1$ to $i = N$). There are several techniques to integrate the weak form in meshless methods [Huang et al., 2019], for example, nodal integration is used in SPH methods.

Instead of accomplishing the discretization process directly by using the definition of a discrete scalar product, Ivanova et al. introduce a set of functions ψ_i linked to each particle i that verifies the Partition of Unity property (PU). Satisfying the PU property means that for any point $\boldsymbol{x}$ in the domain Ω at any instant time t the following identity holds exactly

$$\sum_i \psi\left(\boldsymbol{x} - \boldsymbol{x}_i(t)\right) = 1 \qquad \text{or} \qquad \sum_i \psi_i\left(\boldsymbol{x}\right) = 1 \tag{5.6}$$

By invoking the Fubini's Theorem, the weak form given by Eq. (5.5) is rewritten as

$$\forall \varphi \in C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right) \qquad \int_{\mathbb{R}^+} \int_{\mathbb{R}^d} \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi \right) dx^d \, dt = \mathbf{0} \tag{5.7}$$

Leaving the time integral temporally out of the derivation, attention is set on conducting the spatial discretization of the spatial integral

$$\int_{\mathbb{R}^d} \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi \right) dx^d \tag{5.8}$$

Using the Partition of Unity property (Eq. (5.6)) in Eq. (5.8) allows us to write

$$\int_{\mathbb{R}^d} 1(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi)\, dx^d = \int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x}) \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi\right) dx^d \tag{5.9}$$

where we have decided to emphasize only the argument dependence for the particle functions $\psi_i(\boldsymbol{x})$. After expanding the products, we note that each integral demands a different approximation, and for that reason we identify each integral term with Roman numbers

$$\underbrace{\int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x})\, \boldsymbol{U}\dot{\varphi}\, dx^d}_{I} + \underbrace{\int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x})\, \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}}\, dx^d}_{II} + \underbrace{\int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x})\, \boldsymbol{S}\varphi\, dx^d}_{III} \tag{5.10}$$

Before starting the approximation of each term, we expose some properties associated with the PU. The insertion of the PU property in the integral that gives the volume of the domain yields

$$V = \int_\Omega dx^d = \int_\Omega 1\, dx^d = \int_\Omega \sum_i \psi_i(\boldsymbol{x})\; dx^d = \sum_i \int_\Omega \psi_i(\boldsymbol{x})\; dx^d = \sum_i V_i \tag{5.11}$$

that provides a partition of the domain Ω, by assigning to a generic particle i the volume V_i given by

$$V_i = \int_\Omega \psi_i(\boldsymbol{x})\; dx^d \tag{5.12}$$

This property is of great importance. It shows that an effective volume associated with a particle i is expressed as the integral of the corresponding particle function $\psi_i(\boldsymbol{x})$ over the whole domain. Particles have a compact support associated with the kernel function, but there is overlapping between the support domain of neighboring particles. The overlapping prevents to evaluate the integral over the domain as the sum of the integrals over the support of each particle. Eq. (5.11) and Eq. (5.12) show that by inserting the PU inside the integral, it is possible to express the integral over the domain as the sum of the contributions of the particles. We also remark that Eq. (5.12) is obtained without using a cell or invoking a tessellation procedure. Particle functions $\psi_i(\boldsymbol{x})$ are null outside the boundary domain, and therefore the region of integration $\mathbb{R}^d$ of Eq. (5.10) is replaced by Ω.

Coming back to the development of Eq. (5.8), the term I can be approximated as

$$\begin{aligned} I &= \sum_i \int_\Omega \psi_i(\boldsymbol{x})\, \boldsymbol{U}(\boldsymbol{x})\dot{\varphi}(\boldsymbol{x})\, dx^d \approx \sum_i \int_\Omega \psi_i(\boldsymbol{x})\, \boldsymbol{U}(\boldsymbol{x}_i)\, \dot{\varphi}(\boldsymbol{x}_i)\; dx^d \\ &= \sum_i \boldsymbol{U}(\boldsymbol{x}_i)\, \dot{\varphi}(\boldsymbol{x}_i) \int_\Omega \psi_i(\boldsymbol{x})\, dx^d = \sum_i \boldsymbol{U}(\boldsymbol{x}_i)\, \dot{\varphi}(\boldsymbol{x}_i)\, V_i = \sum_i \boldsymbol{U}_i \dot{\varphi}_i V_i \end{aligned} \tag{5.13}$$

where the dependence of the spatial variable is included to show clearly the one point quadrature rule used for the approximation.

For the term II, the following gradient approximation is introduced $\nabla\varphi(\boldsymbol{x},t) \approx \widehat{\nabla\varphi}(\boldsymbol{x}) = \sum_j \nabla\psi_j(\boldsymbol{x})\,\varphi(\boldsymbol{x}_j) = \sum_j \nabla\psi_j(\boldsymbol{x})\,\varphi_j$ yielding

$$II = \int_\Omega \sum_i \psi_i(\boldsymbol{x})\,\nabla\varphi(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x})\,dx^d \approx \int_\Omega \sum_i \psi_i(\boldsymbol{x})\,\widehat{\nabla\varphi}(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x})\,dx^d$$
$$= \int_\Omega \sum_i \psi_i(\boldsymbol{x}) \sum_j \varphi_j \nabla\psi_j(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x})\,dx^d \quad (5.14)$$

At this moment it is possible to take advantage of other particle function property to obtain a more convenient expression for the term II. By taking the gradient of the PU identity, it can be probed that the gradient of particle functions satisfy the Partition of Nullity (PN)

$$\underbrace{\sum_j \psi_j(\boldsymbol{x}) = 1}_{PU} \quad \underbrace{\sum_j \nabla\psi_j(\boldsymbol{x}) = \mathbf{0}}_{PN} \quad (5.15)$$

By using the PN property we add a term with null contribution to Eq. (5.14) obtaining

$$II \approx \int_\Omega \sum_i \psi_i(\boldsymbol{x}) \sum_j (\varphi_j - \varphi_i)\,\nabla\psi_j(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x})\,dx^d \quad (5.16)$$

By expanding the terms in brackets and moving the particle functions $\psi_i(\boldsymbol{x})$ inside the summation over index j, the term II reads as

$$II \approx \int_\Omega \left(\sum_i \sum_j \varphi_j \psi_i(\boldsymbol{x}) \nabla\psi_j(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) - \sum_i \sum_j \varphi_i \psi_i(\boldsymbol{x}) \nabla\psi_j(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \right) dx^d \quad (5.17)$$

The integrand is composed of two double sums over all particles of the domain. It is possible to interchange index i and j in the first term, obtaining

$$II \approx \int_\Omega \left(\sum_j \sum_i \varphi_i \psi_j(\boldsymbol{x}) \nabla\psi_i(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) - \sum_i \sum_j \varphi_i \psi_i(\boldsymbol{x}) \nabla\psi_j(\boldsymbol{x})\cdot\mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \right) dx^d \quad (5.18)$$

By rearranging terms and using the additive property of the integral, the II term is given by

$$II \approx -\sum_i \sum_j \varphi_i \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \left(\psi_i(\boldsymbol{x})\nabla\psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x})\nabla\psi_i(\boldsymbol{x})\right)\, dx^d \quad (5.19)$$

In order to perform the integration of term III , we proceed in the same manner as for term I, to obtain

$$III = \sum_i \int_\Omega \psi_i(\boldsymbol{x})\, \boldsymbol{S}(\boldsymbol{x}) \varphi(\boldsymbol{x})\, dx^d \approx \sum_i \boldsymbol{S}_i \varphi_i V_i \tag{5.20}$$

Introducing the approximations obtained in Eq. (5.13), Eq. (5.19) and Eq. (5.20) into Eq. (5.7) yields

$$\int_{\mathbb{R}^+} \left(\sum_i \boldsymbol{U}_i \dot{\varphi}_i V_i - \sum_i \sum_j \varphi_i \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \big(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\big) dx^d + \sum_i \boldsymbol{S}_i \varphi_i V_i \right) dt \approx \mathbf{0} \tag{5.21}$$

The overdot operator was used as abbreviation for the advective derivative according to $L^*_{\boldsymbol{w}}(\varphi) = \dot{\varphi} = \partial\varphi/\partial t + \boldsymbol{w} \cdot \nabla\varphi$. Application of integration by parts and noting that the tests functions have compact support (in space and time) the transient term in Eq. (5.21) can be expressed by

$$\sum_i \int_{\mathbb{R}^+} \boldsymbol{U}_i \frac{d\varphi_i}{dt} V_i\, dt = \sum_i \int_{\mathbb{R}^+} \frac{d\left(\boldsymbol{U}_i \varphi_i V_i\right)}{dt} dt - \sum_i \int_{\mathbb{R}^+} \varphi_i \frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} dt = - \int_{\mathbb{R}^+} \left(\sum_i \varphi_i \frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} \right) dt \tag{5.22}$$

Substitution of Eq. (5.22) into Eq. (5.21) gives

$$\int_{\mathbb{R}^+} \left(- \sum_i \varphi_i \frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} - \sum_i \sum_j \varphi_i \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \big(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\big)\, dx^d + \sum_i \boldsymbol{S}_i \varphi_i V_i \right) dt \approx \mathbf{0} \tag{5.23}$$

As the integral must approach zero for any time, the integrand should approach zero

$$\sum_i \varphi_i \left(\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \big(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\big)\, dx^d - \boldsymbol{S}_i V_i \right) \approx \mathbf{0} \tag{5.24}$$

As this relation must hold for any test function, it demands that

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \left(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\right) dx^d \approx \boldsymbol{S}_i V_i \tag{5.25}$$

Now, we note that the second term takes the form of a sum of integrals involving the active particle i with its neighbor particles. The term collects the action of neighbor particles j over the active particle i. We focus the attention on one individual neighbor and we note that particle functions are dimensionless and the gradient of the particle functions have dimensions of $[L]^{-1}$. Moreover, particle functions are scalar and the gradient of particle functions are vectors. The integral obeys the typical expression of a flux crossing a surface. Based on this analogy we represent the interaction flux of particle j over particle i as $(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij}$

$$\int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \left(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\right) dx^d \equiv (\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} \tag{5.26}$$

It is very important to remark that the area $\boldsymbol{A}$ used to denote the term $(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij}$ in Eq. (5.26) is not associated to any geometry entity. Some authors use the term effective area to emphasize the differences with a real area associated with the face of a cell. Using this compact nomenclature Eq. (5.25) can be rewritten as

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j (\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} \approx \boldsymbol{S}_i V_i \tag{5.27}$$

The meshless formulation presented does not provide a full discretization, because the tensor flux has not been discretized and the effective area $\boldsymbol{A}_{ij}$ requires a quadrature rule to obtain a discrete approximation. Moreover, the set of particle functions satisfying the PU and PN are key in the process, but in [Ivanova et al., 2013] nothing is said about its definition. At this point, the discretization is not applicable in practice since it is incomplete. However, it has the advantage of offering a meshless formulation that allows easy comparison with the FVM and traditional SPH method.

5.3.3. SPH discretized equations

The first thing to be clarified is that there is not a unique set of discretized equations for the Euler equations using SPH. In the literature different strategies to discretize the Euler equations have been explored. The simplest procedure starts with the choice of a set of Euler equations in non-conservative form, and then the continuous spatial derivatives are replaced by their discrete kernel approximations. A more elegant procedure to derive the discretized equations adopts the Lagrangian Dynamics. Using the variational principle avoids some ad hoc choices introduced by the simple procedure. Excellent reviews of the SPH procedure are given in the publications by Monaghan [Monaghan, 2005], Price [Price, 2012], Rosswog [Rosswog, 2009], and Springel [Springel, 2010] and Violeau's book [Violeau, 2012].

As SPH is a pure Lagrangian method, the more convenient presentation for the Euler equations is the non-conservative form since material derivatives appear explicitly on the left hand side of the governing equations. For the energy equation we can use the internal energy (e) or the total energy (E) as variable.

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{u} \tag{5.28}$$

$$\frac{d\boldsymbol{u}}{dt} = -\frac{\nabla p}{\rho} + \boldsymbol{f} \tag{5.29}$$

$$\frac{de}{dt} = -\frac{p}{\rho}\nabla \cdot \boldsymbol{u} \qquad \frac{dE}{dt} = -\frac{1}{\rho}\nabla \cdot (p\boldsymbol{u}) + \boldsymbol{f} \cdot \boldsymbol{u} \tag{5.30}$$

Eq. (5.28), Eq. (5.29) and Eq. (5.30) express the non-conservative form for the mass, momentum and energy respectively. In the right hand side of Eq. (5.28) the divergence of the velocity appears meanwhile in Eq. (5.29) the gradient of the pressure is involved. Omitting the discretization of the energy equation we note that the main discretizations invoked for SPH are conducted by one of the two forms presented hereafter.

The traditional SPH discretization was adopted by Monaghan and its the usual set of SPH equations used in Astrophysics. The expressions assume a constant smoothing length for the particles and the semi-discrete corresponding to Eq. (5.28), Eq. (5.29) read as

$$\frac{D\rho_i}{Dt} = \sum_j m_j \left(\boldsymbol{u}_i - \boldsymbol{u}_j\right) \nabla_i W_{ij} \tag{5.31}$$

$$\frac{D\boldsymbol{u}_i}{Dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \nabla_i W_{ij} + \boldsymbol{f}_i \tag{5.32}$$

An alternative SPH discretization was proposed by Colagrossi and Landrini [Colagrossi & Landrini, 2003]. This alternative SPH derivation is the base for the $\delta - SPH$ method and is widely used in ocean engineering.

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_j} \left(\boldsymbol{u}_i - \boldsymbol{u}_j\right) \nabla_i W_{ij} \tag{5.33}$$

$$\frac{D\boldsymbol{u}_i}{Dt} = -\sum_j m_j \left(\frac{p_i + p_j}{\rho_i \rho_j}\right) \nabla_i W_{ij} + \boldsymbol{f}_i \tag{5.34}$$

Both forms of the SPH equations are self-consistent because they were derived invoking the variational principle.

5.3.4. Connections between FVM and MFVM

The similarities between the MFVM and the FVM can be clearly seen by comparison of Eq. (5.2) and Eq. (5.27). The main difference between the formulations resides in the evaluation of the fluxes given by Eq. (5.3) and Eq. (5.26). In FVM fluxes are calculated over a face of a geometric cell meanwhile in the MFVM fluxes are defined over an effective area expressed in terms of particle functions.

The MFVM proposed by Ivanova et al. [Ivanova et al., 2013] does not provide a full discretization of the Euler equations. The set of particle functions are not defined and the flux term expressed by $(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij}$ needs to be evaluated numerically with the information carried by the particles. However, the lack of an approximation for the flux and the effective area allows us to appreciate more clearly the similarity of MFVM with FVM.

5.3.5. Connections between MFVM and SPH

In this subsection, we outline the derivation of the traditional SPH equations and we show how to recover these equations starting from the MFVM given by Eq. (5.27) and Eq. (5.26).

Particle functions $\psi_i(\boldsymbol{x})$ satisfying the PU property plays a key role in the MFVM. On the contrary, the traditional SPH equations are derived in terms of the kernel function $W_i(\boldsymbol{x})$. Although the derivations seem different we show the path to recover the traditional Lagrangian SPH equations starting with the MFVM. The key factor for the analysis is based on the identification of a hidden PU associated with the kernel approximation.

SPH is based on the smoothing of the mass assigned to a set of discrete points. Based on that we define the density spread in an arbitrary point $\boldsymbol{x}$ by the presence of the mass m_i located in point $\boldsymbol{x}_i$ by $\rho_i(\boldsymbol{x}) = m_i W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right)$. The density at point $\boldsymbol{x}$ is obtained by summation over the neighboring particles (n_x) inside the kernel support, which is expressed mathematically by $\rho(\boldsymbol{x}) = \sum_j^{n_x} W\left(\boldsymbol{x} - \boldsymbol{x}_j, h(\boldsymbol{x})\right) m_j$. The following partition of unity associated with SPH can be inferred

$$\psi_i^{SPH}(\boldsymbol{x}) = \frac{m_i W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right)}{\sum_j^{n_x} W\left(\boldsymbol{x} - \boldsymbol{x}_j, h(\boldsymbol{x})\right) m_j} = \frac{\rho_i(\boldsymbol{x})}{\rho(\boldsymbol{x})} \tag{5.35}$$

Before paying attention to the conservative equations, it is illustrative to examine the expression of the effective volume associated with a particle in MFVM

$$V_i = \int_\Omega \psi_i^{SPH}(\boldsymbol{x})\, dx^d = \int_\Omega \frac{m_i}{\rho(\boldsymbol{x})} W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right)\, dx^d \tag{5.36}$$

It is possible to proceed further after approximating the kernel by the delta function, namely $W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) \approx \delta\left(\boldsymbol{x} - \boldsymbol{x}_i\right)$. By means of the integral identity of the delta function the following relation is derived

$$V_i = \int_\Omega \frac{m_i}{\rho(\boldsymbol{x})} W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right)\, dx^d \approx \int_\Omega \frac{m_i}{\rho(\boldsymbol{x})} \delta\left(\boldsymbol{x} - \boldsymbol{x}_i\right)\, dx^d = \frac{m_i}{\rho_i} \tag{5.37}$$

In traditional SPH the mass of the particle is kept fixed, and the density is derived by using the kernel approximation. It is possible to obtain a volume associated with the particle through the density definition, but it is not required for solving the governing equations. On the contrary, in the MFVM the mass of the particle is evolved from the mass conservation (Eq. (5.27) with $U^1 = \rho$) and the particle volume is defined by Eq. (5.11). To obtain a numerical value of the volume we need a quadrature rule to approximate the previous integral. There is also the alternative choice of evolving the volume of the particle using the generalization of the Leibniz rule (see Eq. (4.32)).

For the momentum equation, we start with the general equation of the meshless finite volume formulation given by Eq. (5.27) and we set the particular settings corresponding to SPH. Particles move with the fluid velocity ($\boldsymbol{w} = \boldsymbol{u}$), and the vector of conservative variables $\boldsymbol{U}_i$ and source terms S_i takes the particular form $\boldsymbol{U}_i = \rho_i \boldsymbol{u}_i$ and $S_i = \rho_i \boldsymbol{f}_i$ thus giving

$$\left.\frac{d}{dt}\right|_u \rho_i \boldsymbol{u}_i V_i + \sum_j (\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij} = \rho_i \boldsymbol{f}_i V_i \tag{5.38}$$

We then proceed by computing the flux terms $(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij}$. Because SPH is a Lagrangian method, the flux of the momentum is a diagonal tensor $\mathbf{F}_{\boldsymbol{u}} = p\mathbf{I}$ involving only the pressure term. With these considerations, the general expression for the meshless flux given by Eq. (5.26) reads as

$$(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij} = \int_\Omega \left[\psi_i(\boldsymbol{x}) p(\boldsymbol{x}) \nabla\psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) p(\boldsymbol{x}) \nabla\psi_i(\boldsymbol{x})\right]\, dx^d \tag{5.39}$$

Now, we insert in Eq. (5.39) the expressions for $\psi_i(\boldsymbol{x})$ and $\nabla\psi_i(\boldsymbol{x})$ that correspond to the SPH kernel approximation, that is

$$\psi_i^{SPH}(\boldsymbol{x}) = \frac{m_i}{\rho(\boldsymbol{x})} W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) \qquad \nabla\psi_i^{SPH}(\boldsymbol{x}) \approx \frac{m_i}{\rho(\boldsymbol{x})} \nabla W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) \tag{5.40}$$

resulting in the following expression for the momentum flux $(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij}$

$$\begin{aligned}(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij} \approx \int_\Omega \Bigg[& \frac{m_i}{\rho(\boldsymbol{x})} W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) p(\boldsymbol{x}) \frac{m_j}{\rho(\boldsymbol{x})} \nabla W\left(\boldsymbol{x} - \boldsymbol{x}_j, h(\boldsymbol{x})\right) \\ & - \frac{m_j}{\rho(\boldsymbol{x})} W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) p(\boldsymbol{x}) \frac{m_i}{\rho(\boldsymbol{x})} \nabla W\left(\boldsymbol{x} - \boldsymbol{x}_i, h(\boldsymbol{x})\right) \Bigg]\, dx^d\end{aligned} \tag{5.41}$$

The momentum flux $(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij}$ given by Eq. (5.41) involves a complex integration including the kernel and gradient kernel for particles i and j. However, by shrinking

the kernel in the delta function we can use the properties of the delta function to obtain the approximation

$$(\mathbf{F}_{\boldsymbol{u}} \cdot \boldsymbol{A})_{ij} \approx \frac{m_i m_j}{\rho_i^2} p_i \nabla_i W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h_i\right) - \frac{m_i m_j}{\rho_j^2} p_j \nabla_j W\left(\boldsymbol{x}_j - \boldsymbol{x}_i, h_j\right) \tag{5.42}$$

Replacing Eq. (5.42) into Eq. (5.38) yields

$$\left.\frac{d}{dt}\right|_{\boldsymbol{u}} m_i \boldsymbol{u}_i + \sum_j \left[\frac{m_i m_j}{\rho_i^2} p_i \nabla_i W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h_i\right) - \frac{m_i m_j}{\rho_j^2} p_j \nabla_j W\left(\boldsymbol{x}_j - \boldsymbol{x}_i, h_j\right) \right] = \rho_i \boldsymbol{f}_i V_i \tag{5.43}$$

For a uniform smoothing length $h = h_i = h_j$, we could use the symmetric property of the kernel function

$$\nabla_i W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) = -\nabla_j W\left(\boldsymbol{x}_j - \boldsymbol{x}_i, j\right) \qquad \nabla_i W_{ij} = -\nabla_j W_{ji} \tag{5.44}$$

and after dividing by the constant mass m_i of the particle we arrive at the final expression for the momentum equation given by

$$\left.\frac{d}{dt}\right|_{\boldsymbol{u}} \boldsymbol{u}_i = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} + \boldsymbol{f}_i \tag{5.45}$$

The discrete momentum equation given by Eq. (5.45) is coincident with Eq. (5.32) obtained with the traditional SPH formulation by Monaghan. Thus, it can be concluded that the SPH method can be considered as a particular configuration of the Meshless Finite Volume Method.

In the beginning of the chapter we introduced the idea that MFVM methods were the bridge between mesh-based and meshless methods. Once the discretized equations are analyzed it can be asserted now that it is more appropriate to consider that FVM and SPH are particular methods contained in the MFVM class of methods.

5.4. MLSPH-ALE. A common framework for meshless-fv methods

In the previous section we outlined the relations between mesh-based methods and SPH methods. The hybrid features of meshless-fv methods have attracted the attention of many researchers to develop meshless methods that over-perform the mesh-based and SPH methods in solving a wide variety of applications. As a consequence, several meshless-fv methods were conceived in the last decade. Some of them were developed to solve astrophysical problems ([Hopkins, 2015], [Gaburov & Nitadori, 2011], [Hinz, 2020], [Alonso-Asensio, 2019]) meanwhile other methods were aimed to solve the flow in hydraulic turbomachines ([Marongiu, 2007], [Vessaz, 2015]). Some other meshless-fv

methods like the Finite Volume Particle Method (FVPM) and the SPH-ALE of Vila were conceived as a general method to solve hyperbolic conservation laws.

Meshless-fv methods have not achieved the same degree of maturity as the FEM and FVM, so it seems reasonable that in the current state of development several of the meshless-fv methods cited will evolve and some new meshless formulations appear in a near future. Thus, the list is not exhaustive but it is representative of the different communities that use meshless-fv methods. In the next sections we will prove that the MLSPH-ALE is a general meshless-fv formulation. After a selection of representative meshless-fv in the literature, it is shown that adopting some particular settings in MLSPH-ALE leads to equivalent semi-discrete equations

5.4.1. The Finite Volume Particle Method (FVPM)

The Finite Volume Particle Method (FVPM) was first developed by Hietel, Steiner and Struckmeier in [Hietel et al., 2000] for a system of conservation laws in the spatial domain $\Omega = \mathbb{R}^d$. Then, Junk and Struckmeier [Junk & Struckmeier, 2000] proposed a more stable discretization in a bounded domain $\Omega \subset \mathbb{R}^d$. Keck and Hietel incorporate a projection technique for incompressible flow [Keck & Hietel, 2005]. Lamichanne [Lamichhane, 2001] and Teleaga and Struckmeier [Teleaga, 2005; Teleaga & Struckmeier, 2008] extended the method to moving domains. Quinlan and coworkers have conducted a very active research work on the FVPM on the last decade. Some of their main contributions are the extension of the FVPM to viscous flow [Nestor et al., 2009] and free-surface flows [Quinlan, 2018].

The FVPM is a method that was developed to be a hybrid method with the flexibility of a particle method and the assurance of conservativeness inherent to FVM. The FVPM assigns a particle function to each particle that automatically satisfies the Partition of Unity property. This function is denoted as $\psi_i(\boldsymbol{x}, t)$

$$\psi_i(\boldsymbol{x}, t) \equiv \frac{W(\boldsymbol{x} - \boldsymbol{x}_i, t)}{\sum_{j=1}^{N} W(\boldsymbol{x} - \boldsymbol{x}_j, t)} = \frac{W_i(\boldsymbol{x}, t)}{\sigma(\boldsymbol{x}, t)} \tag{5.46}$$

where the summation over all domain particle functions at point $\boldsymbol{x}$ for an instant time t is designated more compactly as $\sigma(\boldsymbol{x}, t) \equiv \sum_{j=1}^{N} W(\boldsymbol{x} - \boldsymbol{x}_j, t)$. It can be observed that the particle function ψ_i takes the form of a field variable expressed in Eulerian description. Note that $\boldsymbol{x}$ and t are the independent arguments of a variable ψ_i. Therefore, the time derivative of the particle function can be inferred by using the concept of the derivative following a particle as

$$\frac{d}{dt}\psi_i(\boldsymbol{x}, t) = \frac{\partial}{\partial t}\psi_i(\boldsymbol{x}, t) + \boldsymbol{w} \cdot \nabla\psi_i(\boldsymbol{x}, t) \tag{5.47}$$

The expression for the derivative of the particle function is a key element in the procedure to obtain the semi-discrete equations of the FVPM. At this point it is important

to remark some differences between the tasks conducted by $\psi_i(\boldsymbol{x})$ in the MFVM and in the FVPM. Both particle functions are used to conduct the subdivision of the domain. In the FVPM, the $\psi_i(\boldsymbol{x},t)$ functions are also used as the test functions to define the weak form. Meanwhile, in the MFVM the particle functions $\psi_i(\boldsymbol{x})$ were introduced to facilitate the integration of the weak form. The FVPM is aimed to solve general conservation laws, written in the form

$$\frac{\partial}{\partial t}\boldsymbol{U} + \nabla \cdot \mathbf{F}_{\mathrm{E}}(\boldsymbol{U}) = \boldsymbol{S} \quad \forall \boldsymbol{x} \in \Omega(t) \subset \mathbb{R}^d, \quad t > 0 \tag{5.48}$$

with initial conditions $\boldsymbol{U}(\boldsymbol{x},0) = \boldsymbol{U_0}(\boldsymbol{x}), \forall \boldsymbol{x} \in \Omega(t=0)$ and with suitable boundary conditions. In Eq. (5.48), $\Omega(t) \subset \mathbb{R}^d$ is a bounded domain in $\mathbb{R}^d$, $\boldsymbol{U}(\boldsymbol{x},t) \in \mathbb{R}^m, \quad m > 0$ is the vector of conservative variables, and $\mathbf{F}_{\mathrm{E}}(\boldsymbol{U}(\boldsymbol{x},t))$ is the Eulerian Flux tensor.

The procedure to derive the semi-discrete equations of FVPM, starts by testing the conservation law given by Eq. (5.48) against the particle functions $\psi_i(\boldsymbol{x},t)$ given by Eq. (5.46), resulting in the weak form given by

$$\int_{\Omega(t)} \left(\frac{\partial}{\partial t}\boldsymbol{U} + \nabla \cdot \mathbf{F}_{\mathrm{E}}(\boldsymbol{U}) \right) \psi_i(\boldsymbol{x},t)\, dx^d = \int_{\Omega(t)} \boldsymbol{S}\, \psi_i(\boldsymbol{x},t)\, dx^d, \quad i = 1, \ldots, N \tag{5.49}$$

For the volume integral containing the source term, we can associate a local average to each particle i

$$\boldsymbol{S}_i(t) = \frac{1}{V_i(t)} \int_{\Omega(t)} \boldsymbol{S}(\boldsymbol{x},t)\psi_i(\boldsymbol{x},t)\, dx^d \tag{5.50}$$

By performing integration by parts in the transient and flux term and using Eq. (5.50) we obtain

$$\int_{\Omega(t)} \frac{\partial(\boldsymbol{U}\psi_i)}{\partial t} dx^d + \int_{\Omega(t)} \nabla\cdot(\mathbf{F}_{\mathrm{E}}\, \psi_i)\, dx^d = \int_{\Omega(t)} \left(\boldsymbol{U}\frac{\partial \psi_i}{\partial t} + \mathbf{F}_{\mathrm{E}} \cdot \nabla \psi_i \right) dx^d + \boldsymbol{S}_i V_i \tag{5.51}$$

The Leibniz rule of integration enable us to write the following relation

$$\frac{d}{dt}\int_{\Omega(t)} \boldsymbol{U}\psi_i\, dx^d = \int_{\Omega(t)} \frac{\partial(\boldsymbol{U}\psi_i)}{\partial t}\, dx^d + \int_{\partial\Omega(t)} (\boldsymbol{U}\psi_i \otimes \boldsymbol{b}) \cdot \boldsymbol{n}\, dx^{d-1} \tag{5.52}$$

with $\boldsymbol{b}(\boldsymbol{x},t)$ denoting the velocity of the boundary $\partial\Omega(t)$. After application of Leibniz rule, Eq. (5.51) becomes

$$\frac{d}{dt}\int_{\Omega(t)} \boldsymbol{U}\psi_i\, dx^d + \underbrace{\int_{\partial\Omega(t)} \psi_i\, (\mathbf{F}_{\mathrm{E}} - \boldsymbol{U} \otimes \boldsymbol{b}) \cdot \boldsymbol{n}\, dx^{d-1}}_{B_i(x,t)} = \int_{\Omega(t)} \left(\boldsymbol{U}\frac{\partial \psi_i}{\partial t} + \mathbf{F}_{\mathrm{E}} \cdot \nabla \psi_i \right) dx^d + \boldsymbol{S}_i V_i \tag{5.53}$$

We rename the integral boundary term as $B_i(\boldsymbol{x},t)$. This boundary term appears only for particles i which are close to the boundary, i.e. $D_i \cap \partial\Omega \neq 0$. In the following derivation, we neglect this boundary term and we restrict the analysis to interior particles far from the boundary of the domain. To proceed further, we need to obtain a more convenient expression for the transient term on the left hand side and also for the two terms on the right hand side of Eq. (5.53).

In a very similar form to FVM, we can associate a local average to each particle i, through the equation

$$\boldsymbol{U}_i(t) = \frac{1}{V_i(t)} \int_{\Omega(t)} \boldsymbol{U}(\boldsymbol{x},t)\psi_i(\boldsymbol{x},t)\, dx^d \tag{5.54}$$

where $V_i(t)$ is the time-dependent volume of the particle i given by

$$V_i(t) = \int_{\Omega(t)} \psi_i(\boldsymbol{x},t)\, dx^d \tag{5.55}$$

Eq. (5.54) is denoted as the discrete particle property associated with the FVPM. The discrete particle property relation assigns the cell averaged values to each cell. It can be noticed that Eq. (5.55) derives from the discrete particle property associated with a constant unit function. In classical FVM the test functions are taken as the characteristic functions $I_{\Omega_i}(\boldsymbol{x})$ of the control volume Ω_i. A characteristic function is a piecewise constant function that values 1 in the corresponding cell and values 0 otherwise.

By taking the time rate of Eq. (5.54) we obtain the relation

$$\frac{d}{dt} \int_{\Omega(t)} \boldsymbol{U}\psi_i\, dx^d = \frac{d\,(\boldsymbol{U}_i V_i)}{dt} \tag{5.56}$$

where the time rate of a continuous integral over the domain is given in terms of a time rate of a discrete product of particle properties. By using Eq. (5.56) and dropping the boundary term $B_i(\boldsymbol{x},t)$, Eq. (5.53) gives

$$\frac{d\,(\boldsymbol{U}_i V_i)}{dt} = \int_{\Omega(t)} \left(\boldsymbol{U}\frac{\partial \psi_i}{\partial t} + \mathbf{F}_{\mathrm{E}} \cdot \nabla\psi_i \right) dx^d + \boldsymbol{S}_i V_i \tag{5.57}$$

which is valid for particles far away for the boundary $\partial\Omega(t)$. For the particle function $\psi_i(\boldsymbol{x},t)$ defined in Eq. (5.46) it can be proved that (see [Hietel et al., 2000] [3])

$$\frac{\partial \psi_i(\boldsymbol{x},t)}{\partial t} = \sum_{j=1}^{N} \left(\boldsymbol{w}_j(t)\boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t) - \boldsymbol{w}_i(t)\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t) \right) \tag{5.58}$$

$$\nabla\psi_i(\boldsymbol{x},t) = \sum_{j=1}^{N} \left(\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t) - \boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t) \right) \tag{5.59}$$

[3] Hietel et al. denote the transport velocity of the particles $\boldsymbol{w}(t)$ by $\dot{\boldsymbol{x}}(t)$

where $\boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t)$ and $\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t)$ are vector functions localized on the intersection of the supports of particle i and particle j, and are defined by

$$\boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t) \equiv \psi_i(\boldsymbol{x},t)\frac{\nabla W_j(\boldsymbol{x},t)}{\sigma(\boldsymbol{x},t)} \quad \boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t) \equiv \psi_j(\boldsymbol{x},t)\frac{\nabla W_i(\boldsymbol{x},t)}{\sigma(\boldsymbol{x},t)} \tag{5.60}$$

Inserting Eq. (5.58) and Eq. (5.59) into Eq. (5.57) and after rearranging terms we arrive at Eq. (5.61). In this manipulation the property $\mathbf{a}(\mathbf{b}\cdot\mathbf{c}) = (\mathbf{a}\otimes\mathbf{b})\mathbf{c}$ was used.

$$\frac{d\,(\boldsymbol{U}_i V_i)}{dt} = \sum_{j=1}^{N}\int_{\Omega(t)} \left[\left(\mathbf{F}_{\mathrm{E}} - \boldsymbol{U}\otimes\boldsymbol{w}_i(t)\right)\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t) - \left(\mathbf{F}_{\mathrm{E}} - \boldsymbol{U}\otimes\boldsymbol{w}_j(t)\right)\boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t)\right] dx^d + \boldsymbol{S}_i V_i \tag{5.61}$$

At this moment, it is convenient to focus the attention on the flux expressions appearing inside the brackets of Eq. (5.61). To evaluate the interaction of particle i with particle j we need to evaluate the fluxes $\left(\mathbf{F}_{\mathrm{E}} - \boldsymbol{U}\otimes\boldsymbol{w}_i(t)\right)$ and $\left(\mathbf{F}_{\mathrm{E}} - \boldsymbol{U}\otimes\boldsymbol{w}_j(t)\right)$. The Eulerian flux tensor $\mathbf{F}_{\mathrm{E}}(\boldsymbol{U}) = \mathbf{F}_{\mathrm{E}}(\boldsymbol{x},t)$ and the vector of conservative variables $\boldsymbol{U}(\boldsymbol{x},t)$ are continuous functions of space and time. On the other hand, particle velocities $\boldsymbol{w}_i(t)$ and $\boldsymbol{w}_j(t)$ are functions of time. To evaluate the integral of Eq. (5.61), an approximation of $\boldsymbol{U}$ at the overlap of particles i and j is required.

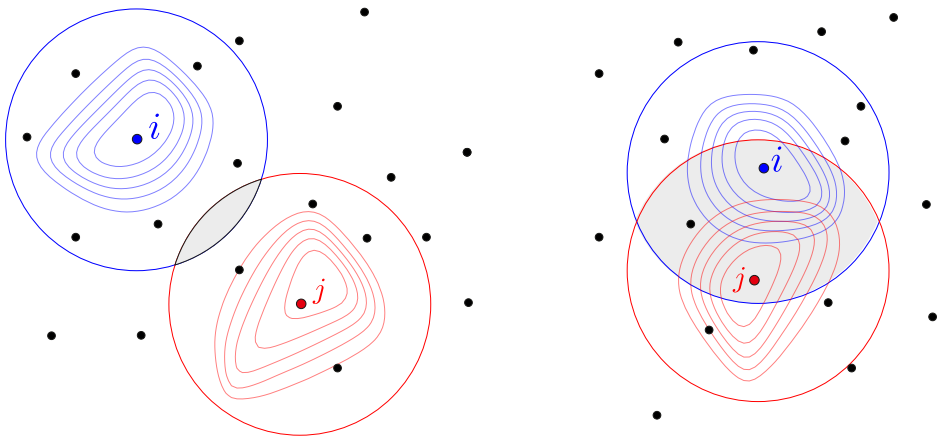


Figure 5.3. Different degree of overlap between particles i and j. Left: Low overlap. Right: High overlap

The approximation invoked to proceed assumes that $\boldsymbol{U}(\boldsymbol{x},t)$ varies little within the overlap of particles i and j, and can be represented by a single value denoted $\overline{\boldsymbol{U}}_{ij}$. Figure 5.3 depicts two different degrees of overlap between particles i and j. The approximation invoked is valid for the situation plotted on the left. The same approximation is adopted for the arbitrary transport velocity. Thus, we denote by $\overline{\boldsymbol{w}}_{ij}$ to the single value transport velocity in the overlap region of particles i and j. The

approximations done renders that both modified fluxes inside brackets of Eq. (5.61) are approximately equal and uniform in each overlap region. Noting that $\boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t)$ and $\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t)$ are zero outside the overlap region the flux can be taken out of the integral giving

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} \approx \sum_{j=1}^{N} \underbrace{\left(\mathbf{F}_{\mathrm{E}}\left(\overline{\boldsymbol{U}}_{ij}\right) - \overline{\boldsymbol{U}}_{ij} \otimes \overline{\boldsymbol{w}}_{ij}\right)}_{I} \underbrace{\int_{\Omega(t)} \left[\boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t) - \boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t)\right] dx^d}_{II} + \boldsymbol{S}_i V_i \quad (5.62)$$

Inspection of Eq. (5.62) shows that after performing the approximations each contributing term of the sum can be expressed as the product of two factors with different nature. The factor inside brackets (I) represents a flux meanwhile the integral (II) contains geometrical information of the neighborhood of particle i.

Once the approximations were introduced, it is observed that flux inside brackets(I) is an advective ALE flux

$$\mathbf{F}_{\boldsymbol{w}}\big(\overline{\boldsymbol{U}}_{ij}, \overline{\boldsymbol{w}}_{ij}\big) \equiv \mathbf{F}_{\mathrm{E}}\big(\overline{\boldsymbol{U}}_{ij}\big) - \overline{\boldsymbol{U}}_{ij} \otimes \overline{\boldsymbol{w}}_{ij} \quad (5.63)$$

Looking for a handier notation for the integral factor the following geometrical coefficients are defined

$$\boldsymbol{\gamma}_{ij}(t) \equiv \int_{\Omega(t)} \boldsymbol{\Gamma}_{ij}(\boldsymbol{x},t)\, \mathrm{d}x^d \quad \boldsymbol{\gamma}_{ji}(t) \equiv \int_{\Omega(t)} \boldsymbol{\Gamma}_{ji}(\boldsymbol{x},t)\, \mathrm{d}x^d \quad (5.64)$$

$$\boldsymbol{A}_{ij}(t) \equiv \boldsymbol{\gamma}_{ij}(t) - \boldsymbol{\gamma}_{ji}(t) \quad (5.65)$$

Substitution of Eq. (5.63), Eq. (5.64) and Eq. (5.65)) into Eq. (5.62) gives the compact semi-discrete form obtained by FVPM

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_{j=1}^{N} \mathbf{F}_{\boldsymbol{w}}\big(\overline{\boldsymbol{U}}_{ij}, \overline{\boldsymbol{w}}_{ij}\big) \boldsymbol{A}_{ij}(t) \approx \boldsymbol{S}_i V_i \quad (5.66)$$

Since we are interested in the unknowns $\boldsymbol{U}_i$, an additional equation for the volumes $V_i(t)$ is needed. We could either evaluate $V_i(t)$ using Eq. (5.55), or update its value by evaluations of its time rate $\dot{V}_i(t)$. By following this second approach, we note that (5.52) express a balance law for a vector of conservative variables $\boldsymbol{U}$. For the particular case of one-component vector, $\boldsymbol{U}$ degenerates in a scalar field U. By setting $U = 1$ the time rate $\dot{V}_i(t)$ obeys

$$\dot{V}_i(t) = \sum_{j=1}^{N} \left(\boldsymbol{\gamma}_{ij}\boldsymbol{w}_j - \boldsymbol{\gamma}_{ji}\boldsymbol{w}_i\right) + \int_{\partial\Omega(t)} \psi_i \boldsymbol{b} \cdot \boldsymbol{n}\, dx^{d-1} \quad (5.67)$$

Eq. (5.66) and Eq. (5.67) supplemented with the initial conditions give the semi-discrete form of the FVPM. This is the most convenient form to compare with other meshless formulations.

The final semi-discrete form given by Eq. (5.66) demands the flux tensor $\mathbf{F}_{\boldsymbol{w}}\left(\overline{\boldsymbol{U}}_{ij}, \overline{\boldsymbol{w}}_{ij}\right)$ to be given in terms of the data stored in the particles. The ALE flux $\mathbf{F}_{\boldsymbol{w}}$ is defined in terms of an uniform representative state at the overlap region of particles i and j. This ALE flux $\mathbf{F}_{\boldsymbol{w}}\left(\overline{\boldsymbol{U}}_{ij}, \overline{\boldsymbol{w}}_{ij}\right)$ is approximated with a numerical ALE flux function $\mathbf{G}_{\boldsymbol{w}}\left(\boldsymbol{U}_i, \boldsymbol{U}_j, \boldsymbol{w}_i, \boldsymbol{w}_j\right)$, as follows

$$\mathbf{F}_{\boldsymbol{w}}\left(\overline{\boldsymbol{U}}_{ij}, \overline{\boldsymbol{w}}_{ij}\right) \equiv \mathbf{F}_{\mathrm{E}}\left(\overline{\boldsymbol{U}}_{ij}\right) - \overline{\boldsymbol{U}}_{ij} \otimes \overline{\boldsymbol{w}}_{ij} \approx \mathbf{G}_{\boldsymbol{w}}\left(\boldsymbol{U}_i, \boldsymbol{U}_j, \boldsymbol{w}_i, \boldsymbol{w}_j\right) = \mathbf{G}\left(\boldsymbol{U}_i, \boldsymbol{U}_j\right) - \overline{\boldsymbol{U}}_{ij} \otimes \overline{\boldsymbol{w}}_{ij} \tag{5.68}$$

where the Eulerian flux $\mathbf{F}_{\mathrm{E}}\left(\overline{\boldsymbol{U}}_{ij}\right)$ is approximated with a numerical flux function $\mathbf{G}\left(\boldsymbol{U}_i, \boldsymbol{U}_j\right)$

$$\mathbf{F}_{\mathrm{E}}\left(\overline{\boldsymbol{U}}_{ij}\right) \approx \mathbf{G}\left(\boldsymbol{U}_i, \boldsymbol{U}_j\right) \tag{5.69}$$

Relations between FVPM and MLSPH-ALE

There are differences in the derivation procedures to obtain the semi-discrete form of the FVPM and MLSPH-ALE. The derivation of the FVPM was started from the strong form of the conservation law in Eulerian form (Eq. (5.48)) meanwhile the MLSPH-ALE started from the conservation law in ALE form (Eq. (4.14)). The FVPM and the MLSPH-ALE also introduce their approximations at different stages in the procedure. FVPM uses particle functions $\psi_i(\boldsymbol{x}, t)$ meanwhile MLSPH-ALE uses particle functions based on MLS $N_i(\boldsymbol{x}, t)$. When the base selected for MLS comprises only the constant term MLS shape functions are identical to the particle functions $\psi_i(\boldsymbol{x}, t)$ used by FVPM. Both formulations share in common that both set of particle functions satisfy the PU property. The FVPM defines an effective surface interaction area $\boldsymbol{A}_{ij}(t)$ that can be integrated using different quadrature rules [Quinlan et al., 2014]. However, it is possible to obtain some relations between these two methods by analysis of their respective semi-discrete forms.

The semi-discrete form for the FVPM was given in Eq. (5.66). The effective interaction area $\boldsymbol{A}_{ij}(t)$ between particle i and j can be expressed using definitions given in Eq. (5.65), Eq. (5.64) and Eq. (5.60) by

$$\boldsymbol{A}_{ij}(t) = \int_{\Omega(t)} \left(\psi_i(\boldsymbol{x}, t) \frac{\nabla W_j(\boldsymbol{x}, t)}{\sigma(\boldsymbol{x}, t)} - \psi_j(\boldsymbol{x}, t) \frac{\nabla W_i(\boldsymbol{x}, t)}{\sigma(\boldsymbol{x}, t)} \right) \mathrm{d}x^d \tag{5.70}$$

where the particle function $\psi_i\left(\boldsymbol{x}, t\right)$ was defined in Eq. (5.46).

The area $\boldsymbol{A}_{ij}(t)$ in Eq. (5.70) is given in terms of particle functions $\psi(\boldsymbol{x}, t)$ and kernel gradients $\nabla W(\boldsymbol{x}, t)$ of the interacting particles. In order to compare with MLSPH-ALE formulation we look for an alternative expression for Eq. (5.70) that is given in terms of particle functions $\psi(\boldsymbol{x}, t)$ and the gradient of particle functions $\nabla\psi(\boldsymbol{x}, t)$. By taking the gradient of Eq. (5.46) which defines the particle function $\psi_i(\boldsymbol{x}, t)$ associated to particle i we obtain

$$\nabla\psi_i(\boldsymbol{x},t) = \frac{\nabla W_i\,(\boldsymbol{x},t)}{\sigma(\boldsymbol{x},t)} + W_i(\boldsymbol{x},t)\nabla\left(\frac{1}{\sigma(\boldsymbol{x},t)}\right) = \frac{\nabla W_i\,(\boldsymbol{x},t)}{\sigma(\boldsymbol{x},t)} - \psi_i(\boldsymbol{x},t)\frac{\nabla\sigma(\boldsymbol{x},t)}{\sigma(\boldsymbol{x},t)} \tag{5.71}$$

where $\nabla W_i(\boldsymbol{x},t)$ can be isolated to obtain

$$\nabla W_i(\boldsymbol{x},t) = \nabla\psi_i(\boldsymbol{x},t)\sigma(\boldsymbol{x},t) + \psi_i(\boldsymbol{x},t)\nabla\sigma(\boldsymbol{x},t) \tag{5.72}$$

Similarly, repeating the operations involved in Eq. (5.71) and Eq. (5.72) for a particle j we obtain an equivalent expression for $\nabla W_j(\boldsymbol{x},t)$

$$\nabla W_j(\boldsymbol{x},t) = \nabla\psi_j(\boldsymbol{x},t)\sigma(\boldsymbol{x},t) + \psi_j(\boldsymbol{x},t)\nabla\sigma(\boldsymbol{x},t) \tag{5.73}$$

By inserting expressions Eq. (5.72) and Eq. (5.73) into Eq. (5.70) the terms involving products $\psi_i(\boldsymbol{x},t)\psi_j(\boldsymbol{x},t)$ are canceled, obtaining an expression for $\boldsymbol{A}_{ij}(t)$ in terms of particle functions, that reads as

$$\boldsymbol{A}_{ij}(t) = \int_{\Omega(t)} \big(\psi_i(\boldsymbol{x},t)\nabla\psi_j(\boldsymbol{x},t) - \psi_j(\boldsymbol{x},t)\nabla\psi_i(\boldsymbol{x},t)\big)\,dx^d \tag{5.74}$$

Eq. (5.74) provides an alternative presentation for the interaction area $\boldsymbol{A}_{ij}$ of FVPM that has a great similarity with the interaction area for MFVM given in Eq. (5.26).

Now, we recall Eq. (4.22) obtained during the MLSPH-ALE derivation process. Since MLSPH-ALE was derived for the Navier-Stokes equations, the ALE flux tensor $\mathbf{F}_{\boldsymbol{w}}$ and the diffuse tensor $\mathbf{D}$ were grouped in a flux tensor $\mathbf{F}$ defined by $\mathbf{F} \equiv \mathbf{F}_{\boldsymbol{w}} - \mathbf{D}$. Thus, for the Euler equations $\mathbf{F} = \mathbf{F}_{\boldsymbol{w}}$, and the MLSPH-ALE method provides the following discretization

$$\frac{d(V_i\boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i}\left[\frac{1}{2}\big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\big) - (\mathbf{F}_{\boldsymbol{w}})_i\right]\left(\int_{\partial\Omega} N_i\,N_j\cdot\boldsymbol{n}\,dx^{d-1}\right.$$
$$\left. - \int_{\Omega} N_j\,\cdot\nabla N_i\,dx^d + \int_{\Omega} N_i\,\cdot\nabla N_j\,dx^d\right) = V_i\,\boldsymbol{S}_i \tag{5.75}$$

Dropping the boundary term $\left(\int_{\partial\Omega} N_i\,N_j\cdot\boldsymbol{n}\,dx^{d-1}\right)$ we note that the MLSPH-ALE method uses an interparticle area given by

$$\boldsymbol{A}_{ij} = \int_{\Omega} N_i\,\cdot\nabla N_j\,dx^d - \int_{\Omega} N_j\,\cdot\nabla N_i\,dx^d \tag{5.76}$$

By comparison of Eq. (5.74) and Eq. (5.76) it can be observed that the effective areas resulting from both methods adopt the same structure. The only difference is that $\boldsymbol{A}_{ij}$ for FVPM (Eq. (5.74)) is given in terms of $\psi_i(\boldsymbol{x},t)$, meanwhile $\boldsymbol{A}_{ij}$ for MLSPH-ALE (Eq. (5.76)) is given in terms of $N_i(\boldsymbol{x},t)$. Since $N_i(\boldsymbol{x},t)$ can be obtained with MLS approximation using any set of polynomial basis functions and $\psi_i(\boldsymbol{x},t)$ correspond to a

particular MLS approximation with constant basis, it can be stated that the expression for $\boldsymbol{A}_{ij}$ obtained with the MLSPH-ALE method is more general than the expression obtained with the FVPM method.

FVPM and MLSPH-ALE method use different numerical approximations to compute their effective areas. MLSPH-ALE method approximates the effective area Eq. (5.76) by using a one point quadrature in the overlap region, resulting in an numerical approximation for $\boldsymbol{A}_{ij}$

$$\boldsymbol{A}_{ij} \approx (V_i \nabla N_{ji} - V_j \nabla N_{ij}) \tag{5.77}$$

Publications dealing with FVPM in the literature have used different quadrature rules to perform the integration of the effective area $\boldsymbol{A}_{ij}$. Numerical approximation of this integral has received great attention by researchers using this method [Quinlan et al., 2014]. Some algorithms build the particle shape functions using kernel with either circular [Jahanbakhsh et al., 2017] or squared support [Jahanbakhsh et al., 2016]. Quinlan et al. [Quinlan & Lobovský, 2018] have shown some preference for hat-shaped kernel functions over the more traditional Gaussian like kernel functions.

5.4.2. SPH-ALE method

The exposition is introduced by recalling the SPH-ALE method of Vila described in Chapter 2. The semi-discrete form of the SPH-ALE method is given by Eq. (2.90), and read as

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i \sum_{j=1}^{n_i} V_j \left((\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_i + (\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_j \right) \frac{\partial}{\partial x^{\alpha}_i} W_{ij} = \boldsymbol{S}_i V_i \tag{5.78}$$

where the fluxes in ALE form were introduced as vectors F^{α}_{w} with the subindex $\boldsymbol{w}$ remarking the ALE character of the flux with respect a transport velocity field. To ease the comparison with other meshless formulations the ALE flux will be expressed as a tensor, reading as

$$\frac{d}{dt}\left(\boldsymbol{U}_i V_i\right) + V_i \sum_{j=1}^{n_i} V_j \left((\mathbf{F}_{\boldsymbol{w}})_i + (\mathbf{F}_{\boldsymbol{w}})_j \right) \nabla_i W_{ij} = \boldsymbol{S}_i V_i \tag{5.79}$$

Obtaining the SPH-ALE formulation from MLSPH-ALE discretization

Recovering the SPH-ALE method from the MLSPH-ALE requires to relate the gradient of the particle shape function ∇N with the gradient of the kernel ∇W.

The semi-discrete-form obtained with MLSPH-ALE method for the Euler equations far from the boundary was given in Eq. (4.25) and it is recalled here

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2} \Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i \Big) - (\mathbf{F}_{\boldsymbol{w}})_i \right] (-V_j \nabla N_{ij} + V_i \nabla N_{ji}) = V_i \boldsymbol{S}_i \tag{5.80}$$

In Table 5.1 the expressions for the kernel and MLS approximations for the function and the gradient of a scalar function u are given

Kernel approximation	MLS approximation
$\langle u(\boldsymbol{x})\rangle = \sum_j u(\boldsymbol{x}_j)\, W(\boldsymbol{x}-\boldsymbol{x}_j, h)\, V_j$	$\hat{u}(\boldsymbol{x}) = \sum_j N(\boldsymbol{x}-\boldsymbol{x}_j, h)\, u(\boldsymbol{x}_j)$
$\langle \nabla u(\boldsymbol{x})\rangle = \sum_j u(\boldsymbol{x}_j)\, \nabla W(\boldsymbol{x}-\boldsymbol{x}_j, h)\, V_j$	$\widehat{\nabla u}(\boldsymbol{x}) = \sum_j \nabla N(\boldsymbol{x}-\boldsymbol{x}_j, h)\, u(\boldsymbol{x}_j)$

Table 5.1. Kernel and MLS approximations for a scalar function and its gradient

Introducing the approximation $\langle \nabla u(\boldsymbol{x})\rangle \approx \widehat{\nabla u}(\boldsymbol{x})$, we obtain the following relation

$$\nabla W(\boldsymbol{x}-\boldsymbol{x}_j, h)\,\omega_j \approx \nabla N(\boldsymbol{x}-\boldsymbol{x}_j, h) \tag{5.81}$$

where the kernel and particle shape functions are both centered at $\boldsymbol{x}_j$, and the gradients are taken with respect to the variable $\boldsymbol{x}$. If now the gradients are evaluated at point $\boldsymbol{x}_i$, we obtain the relation

$$\nabla_i W_{ij}\, V_j \equiv \nabla_i W(\boldsymbol{x}_i-\boldsymbol{x}_j, h)\, V_j \approx \nabla N(\boldsymbol{x}_i-\boldsymbol{x}_j, h) \equiv \nabla N_{ji} \tag{5.82}$$

In Eq. (5.82) we have introduced the concise nomenclature that is used for the SPH kernel gradient and for particle shape functions. Note that the common nomenclature for the kernel gradient introduces a particle subindex in the nabla operator remarking that the gradient is taken with respect to the spatial coordinates of that particle. Similarly, if the kernel/shape function is centered in particle i and evaluate its gradient in the position of particle j we obtain

$$\nabla_j W_{ji}\, V_i \equiv \nabla_j W(\boldsymbol{x}_j-\boldsymbol{x}_i, h)\, V_j \approx \nabla N(\boldsymbol{x}_j-\boldsymbol{x}_i, h) \equiv \nabla N_{ij} \tag{5.83}$$

We can insert Eq. (5.82) and Eq. (5.83) in Eq. (5.80) to obtain

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + V_i \sum_{j=1}^{n_i} V_j \left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right](-\nabla_j W_{ji} + \nabla_i W_{ij}) = V_i\, \boldsymbol{S}_i \tag{5.84}$$

Finally, the symmetry property of the kernel $\nabla_j W_{ji} = -\nabla_i W_{ij}$ enable us to write

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + V_i \sum_{j=1}^{n_i} 2V_j \left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right]\nabla_i W_{ij} = V_i\, \boldsymbol{S}_i \tag{5.85}$$

Comparison of Eq. (5.85) with Eq. (5.79) shows that the MLSPH-ALE method can provide the same semi-discrete form of SPH-ALE method if the interaction flux is given by $\left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big)\right]$ instead of $\left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right]$. Since this self-contribution of particle i could be incorporated or suppressed in the MLSPH-ALE procedure it has been proved that the SPH-ALE method is contained in the MLSPH-ALE method.

An interesting analysis is related with the computational cost and accuracy associated with both formulations. MLSPH-ALE demands the calculation of gradients ∇N_{ji}, meanwhile SPH-ALE requires the calculation of the terms $\nabla_i W_{ij} V_j$. The calculation of ∇N_{ji} depends on the number of points considered for the support domain of the particle and the dimension of the base of polynomials considered for the MLS approximation. Evaluation of $V_j \nabla_i W_{ij}$ only involves the data associated with particle i and j. However, in terms of accuracy, the SPH-ALE method can not provide a correct gradient approximation for a linear field. In order to circumvent this flaw, SPH-ALE needs a renormalization correction, which also demands the solution of a linear system of equations for each particle, as the MLS approximation does.

5.4.3. The Meshless Finite Volume Method (MFVM)

The semi-discrete form of the MFVM proposed by Ivanova et al. [Ivanova et al., 2013] is given by Eq. (5.27)

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j (\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} \approx \boldsymbol{S}_i V_i$$

where the interaction term of particle i over j, denoted as $(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij}$, is defined by Eq. (5.26)

$$(\mathbf{F}_{\boldsymbol{w}} \cdot \boldsymbol{A})_{ij} \equiv \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \left(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\right) dx^d$$

The MFVM does not provide a full discretization for the flux between particles and for the effective area because neither of them are given in terms of data contained in the set of particles. This incomplete discretization has several implications. On one hand, without additional discretization the method is unable to run simulations. On the other hand, the method enjoys general features that are useful to appreciate approximations invoked by other meshless-fv methods.

Relations between the MFVM and the MLSPH-ALE method

The semi-discrete system obtained with MFVM for the Euler equations is given by Eq. (5.25), reading as

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j \int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \left(\psi_i(\boldsymbol{x}) \nabla \psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x}) \nabla \psi_i(\boldsymbol{x})\right) dx^d \approx \boldsymbol{S}_i V_i$$

The semi-discrete system obtained with MLSPH-ALE method for the Navier Stokes equations is given by Eq. (4.22). Particularized for the Euler equations ($\boldsymbol{F} = \boldsymbol{F}_{\boldsymbol{w}}$) and omitting the boundary term we obtain

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right] \left(\int_\Omega N_i \cdot \nabla N_j \, dx^d - \int_\Omega N_j \cdot \nabla N_i \, dx^d\right) = V_i \boldsymbol{S}_i$$

Noting that MFVM and MLSPH-ALE semi-discrete forms only differ in their expressions for the interacting term between particles, we equalize them to establish relations between both methods.

$$\underbrace{\int_\Omega \mathbf{F}_{\boldsymbol{w}}(\boldsymbol{x}) \cdot \big(\psi_i(\boldsymbol{x})\nabla\psi_j(\boldsymbol{x}) - \psi_j(\boldsymbol{x})\nabla\psi_i(\boldsymbol{x})\big)\, dx^d}_{MFVM} \approx \underbrace{\left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right] \left(\int_\Omega N_i \cdot \nabla N_j \, dx^d - \int_\Omega N_j \cdot \nabla N_i \, dx^d\right)}_{MLSPH-ALE} \quad (5.86)$$

Eq. (5.86) shows that the MLSPH-ALE method assumes a constant flux tensor in the overlap region, which can be taken out of the integral. Both formulations share the same form for the effective interaction area with the only exception that MFVM is given in terms of $\psi(\boldsymbol{x})$ and MLSPH-ALE uses $N_i(\boldsymbol{x})$ obtained with MLS. In MLSPH-ALE, a full discretization of the flux is provided since the tensor flux is given in terms of particle data.

5.4.4. The Gizmo-MFV method

We outline here the Meshless Finite Volume (MFV) method in the form given by Hopkins [Hopkins, 2015]. The publication is associated with the open source project Gizmo, and therefore we use the name Gizmo-MFV to refer to this method. In [Hopkins, 2015], the author declares that the derivation of the method it is inspired in the work by Gaburov and Nitadori [Gaburov & Nitadori, 2011]. The derivation procedure by Gaburov and Nitadori was presented for a scalar conservation law meanwhile the procedure presented by Hopkins is presented for the Euler equations. Gaburov and Nitadori consider that their derivation is heuristic. Apart from the main publication linked to the Gizmo-MFV we cite two master thesis that provide a concise introduction and details of the code implementation ([Hinz, 2020], [Alonso-Asensio, 2019]).

The derivation of the Gizmo-MFV method is based on the weak form of the Euler equations presented in the form given by Eq. (2.74). We recall that this weak form was also the base for the MFVM (see 5.3.2). Gizmo-MFV and MFVM methods exploit the PU property of particle shape functions and for that reason, it is illustrative to understand the point at which both formulations separate from each other. In the first

steps of the MFVM derivation, the weak form after application of the Fubini theorem was given by Eq. (5.5) and recalled here, reading as

$$\forall \varphi \in C_0^2\left(\mathbb{R}^d \times \mathbb{R}^{+,*}\right) \quad \int_{\mathbb{R}^+}\int_{\mathbb{R}^d} \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi\right) dx^d dt = 0$$

Then, the PU property is invoked to facilitate the integration in the spatial domain. Since $\sum_i \psi_i(\boldsymbol{x}) = 1$ the following identity holds

$$\int_{\mathbb{R}^d} 1(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi)\, dx^d = \int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x}) \left(\boldsymbol{U}\dot{\varphi} + \nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}} + \boldsymbol{S}\varphi\right) dx^d \tag{5.87}$$

Arrived at this point, the MFVM and the Gizmo-MFV follow different paths. In MFVM the integral containing the flux term $\nabla\varphi \cdot \mathbf{F}_{\boldsymbol{w}}$ is approximated in a different form that integrals containing the transient term $\boldsymbol{U}\dot{\varphi}$ and source term $\boldsymbol{S}\varphi$. In the Gizmo-MFV the three integral terms are approximated identically. The integral approximation adopted for a arbitrary scalar function $f(\boldsymbol{x})$ is given by

$$\begin{aligned}\int_{\mathbb{R}^d} \sum_i \psi_i(\boldsymbol{x})\, f(\boldsymbol{x})\, dx^d = \sum_i \int_\Omega \psi_i(\boldsymbol{x})\, f(\boldsymbol{x})\, dx^d \approx \sum_i \int_\Omega \psi_i(\boldsymbol{x})\, f(\boldsymbol{x}_i)\, dx^d = \\ \sum_i f(\boldsymbol{x}_i) \int_\Omega \psi_i(\boldsymbol{x})\, dx^d = \sum_i f(\boldsymbol{x}_i) V_i = \sum_i f_i V_i\end{aligned} \tag{5.88}$$

where the definition of the volume associated to particle $V_i = \int_\Omega \psi_i(\boldsymbol{x})\, dx^d$ was used.

Application of the discretization given by Eq. (5.88) to Eq. (5.87) leads to

$$\int_{\mathbb{R}^+} \left[\sum_i \left(V_i \boldsymbol{U}_i \dot{\varphi}_i + V_i (\nabla\varphi)_i \cdot (\mathbf{F}_{\boldsymbol{w}})_i + V_i \boldsymbol{S}_i \varphi_i\right)\right] dt \approx \mathbf{0} \tag{5.89}$$

To proceed further, the gradient $(\nabla\varphi)_i$ needs to be replaced by a discrete version $(D_h\varphi)_i$. Gizmo-MFV estimates this gradient using the renormalized gradient approximation following the recommendations given by Lanson and Vila [Lanson & Vila, 2008a]. The renormalization process assures that the discrete approximation reproduces exactly a linear field function. Hopkins [Hopkins, 2015] and Gaburov and Nitadori [Gaburov & Nitadori, 2011] use index notation to express the discrete renormalized gradient approximation by

$$(\nabla\varphi)_i^\alpha \approx (D_h^\alpha \varphi)_i = \sum_j (\varphi_j - \varphi_i)\, \tilde{\psi}_j^\alpha(\boldsymbol{x}_i) \tag{5.90}$$

where $\tilde{\psi}_j^\alpha(\boldsymbol{x}_i)$ are defined in terms of the renormalization matrix $(B_H^{\alpha,\beta})_i$, particle positions and particle function $\psi_j(\boldsymbol{x}_i)$ by

$$\tilde{\psi}_j^\alpha(\boldsymbol{x}_i) \equiv \left(B_H^{\alpha,\beta}\right)_i \left(x_j^\beta - x_i^\beta\right) \psi_j(\boldsymbol{x}_i) \tag{5.91}$$

The renormalization matrix $(B_H^{\alpha,\beta})_i$ is required to satisfy condition

$$\delta^{\alpha,\gamma} = \sum_j \left(B_H^{\alpha,\beta}\right)_i \left(x_j^\beta - x_i^\beta\right)\left(x_j^\gamma - x_i^\gamma\right)\psi_j\left(\boldsymbol{x}_i\right) \tag{5.92}$$

where $\delta^{\alpha,\gamma}$ is the Kronecker delta function. Since the definition of the renormalization matrix $(B_H^{\alpha,\beta})_i$ given by Eq. (5.92) is taken from Hopkins [Hopkins, 2015] we append the subindex H in the designation of the matrix.

After introducing Eq. (5.90) into Eq. (5.89) we arrive at

$$\int_{\mathbb{R}^+} \left[\sum_i \left(V_i \boldsymbol{U}_i \dot{\varphi}_i + V_i \sum_j \left(\varphi_j - \varphi_i\right) \tilde{\psi}_j\left(\boldsymbol{x}_i\right) \cdot \left(\mathbf{F}_{\boldsymbol{w}}\right)_i + V_i \boldsymbol{S}_i \varphi_i \right) \right] dt \approx \mathbf{0} \tag{5.93}$$

We note that discrete values of the test function (φ) appear in Eq. (5.93) in the form of a time derivative $\dot{\varphi}_i$ in the first term, as a difference $(\varphi_j - \varphi_i)$ in second term and as φ_i in the third term. Since the aim of this procedure is to obtain a discrete solution of the weak form, we need to rearrange terms in order to verify Eq. (5.93) for any values of (φ).

Application of integration by parts (See Eq. (5.22)) provides an alternative expression for the first term in Eq. (5.93).

$$\int_{\mathbb{R}^+} \left(\sum_i \boldsymbol{U}_i \frac{d\varphi_i}{dt} V_i \, dt \right) dt = -\int_{\mathbb{R}^+} \left(\sum_i \varphi_i \frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} \right) dt \tag{5.94}$$

For the difference term in Eq. (5.93), the factorization is based on the fact that a double sum is not affected by the renumeration of index

$$\begin{aligned} \sum_i V_i \sum_j \left(\varphi_j - \varphi_i\right) \tilde{\psi}_j^\alpha\left(\boldsymbol{x}_i\right) \left(\boldsymbol{F}_{\boldsymbol{w}}^\alpha\right)_i = \\ \underbrace{\sum_i \sum_j V_i \varphi_j \tilde{\psi}_j^\alpha\left(\boldsymbol{x}_i\right) \left(\boldsymbol{F}_{\boldsymbol{w}}^\alpha\right)_i}_{i \rightleftarrows j} - \sum_i \sum_j V_i \varphi_i \tilde{\psi}_j^\alpha\left(\boldsymbol{x}_i\right) \left(\boldsymbol{F}_{\boldsymbol{w}}^\alpha\right)_i = \\ - \sum_i \varphi_i \sum_j \left[V_i (\boldsymbol{F}_{\boldsymbol{w}}^\alpha)_i \tilde{\psi}_j^\alpha\left(\boldsymbol{x}_i\right) - V_j (\boldsymbol{F}_{\boldsymbol{w}}^\alpha)_j \tilde{\psi}_i^\alpha\left(\boldsymbol{x}_j\right) \right] \end{aligned} \tag{5.95}$$

Replacing Eq. (5.94) and Eq. (5.95) into Eq. (5.93) yields

$$\int_{\mathbb{R}^+} \left[-\sum_i \varphi_i \left(\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j \left[V_i (\boldsymbol{F}_{\boldsymbol{w}}^\alpha)_i \tilde{\psi}_j^\alpha\left(\boldsymbol{x}_i\right) - V_j (\boldsymbol{F}_{\boldsymbol{w}}^\alpha)_j \tilde{\psi}_i^\alpha\left(\boldsymbol{x}_j\right) \right] - V_i \boldsymbol{S}_i \right) \right] dt \approx \mathbf{0} \tag{5.96}$$

This must hold for an arbitrary test function φ_i with $i = 1, ..N$; therefore, the expression inside the parenthesis must vanish, i.e.

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j \left[V_i (\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_i \tilde{\psi}^{\alpha}_j\left(\boldsymbol{x}_i\right) - V_j (\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_j \tilde{\psi}^{\alpha}_i\left(\boldsymbol{x}_j\right)\right] = V_i \boldsymbol{S}_i \tag{5.97}$$

At this step in the derivation, Gizmo-MFV replaces the particle fluxes $(\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_i$ and $(\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_i$ with the numerical flux solution of the moving Riemann problem defined by the states of particles/cells i and j. Defining such a flux as $(\boldsymbol{G}^{\alpha}_{\boldsymbol{w}})_{ij}$, Eq. (5.97) is reinterpreted as

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j (\mathbf{G}_{\boldsymbol{w}})_{ij} \cdot \left[V_i \tilde{\psi}_j\left(\boldsymbol{x}_i\right) - V_j \tilde{\psi}_i\left(\boldsymbol{x}_j\right)\right] = V_i \boldsymbol{S}_i \tag{5.98}$$

where we have changed from the index notation for the spatial components to the more compact vector notation.

Transition from Eq. (5.97) to Eq. (5.98) is not supported by mathematical operations. The formal pass from Eq. (5.97) to Eq. (5.98) demands that $(\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_i = (\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_j = (\boldsymbol{F}^{\alpha}_{\boldsymbol{w}})_{ij}$ and this condition is only verified for uniform flows. [Hopkins, 2015; Gaburov & Nitadori, 2011] explain that the introduction of the solution of the Riemann problem is invoked to automatically include the dissipation terms to stabilize a central scheme.

The semi-discrete form of the Gizmo-MFV method is given by Eq. (5.98). Comparing this equation with Eq.(5.27) it can be inferred the definition of an effective interaction area $A_{ij}(t)$, between particles i and j, which is given by

$$A_{ij} \equiv V_i \tilde{\psi}_j\left(\boldsymbol{x}_i\right) - V_j \tilde{\psi}_i\left(\boldsymbol{x}_j\right) \tag{5.99}$$

Obtaining Gizmo-MFV from MLSPH-ALE

Gizmo-MFV discretization is given by Eq. (5.98)

$$\frac{d\left(\boldsymbol{U}_i V_i\right)}{dt} + \sum_j (\mathbf{G}_{\boldsymbol{w}})_{ij} \cdot \left[V_i \tilde{\psi}_j\left(\boldsymbol{x}_i\right) - V_j \tilde{\psi}_i\left(\boldsymbol{x}_j\right)\right] = V_i \boldsymbol{S}_i$$

whereas the discretization of the MLSPH-ALE method, particularized for the Euler equations ($\boldsymbol{F} = \boldsymbol{F}_{\boldsymbol{w}}$) is given in Eq. (4.25)

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i} \left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right] (-V_j \nabla N_{ij} + V_i \nabla N_{ji}) = V_i \boldsymbol{S}_i \tag{5.100}$$

By identification of the effective fluxes and interaction areas we can obtain the particular settings of the MLSPH-ALE formulation to recover the Gizmo-MFV.

The effective flux of Gizmo-MFV is given by $(\mathbf{G}_{\boldsymbol{w}})_{ij}$ meanwhile in MLSPH-ALE reads $\left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right]$. Hopkins introduced the solution of a Riemann problem expressed by $(\mathbf{G}_{\boldsymbol{w}})_{ij}$. In MLSPH-ALE the derivation procedure yields a central flux expression $\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big)$. For stability reasons the central flux is replaced by the solution of the Riemann problem $\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) \approx (\mathbf{G}_{\boldsymbol{w}})_{ij}$. Therefore, the

only difference in the flux expression resides in the subtraction of flux of particle $(\mathbf{F}_{\boldsymbol{w}})_i$ and this term can optionally be incorporated or suppressed in the MLSPH-ALE procedure.

The condition for equivalent effective interaction areas demands a particular setting in the MLS approximation giving

$$\nabla N\left(\boldsymbol{x}_i-\boldsymbol{x}_j,h\right)=\nabla N_{ji}=\tilde{\psi}_j^{\alpha}\left(\boldsymbol{x}_i\right) \tag{5.101}$$

The $\tilde{\psi}_j^{\alpha}\left(\boldsymbol{x}_i\right)$ were defined in Eq (5.91) to provide a compact expression for the renormalized approximation for the gradient of a field (see Eq. (5.90)).

We now look for the corresponding approximation for the gradient of a field but using MLS approximation technique. We start by expressing the MLS approximation of scalar field φ as

$$\hat{\varphi}(\boldsymbol{x})=\sum_j N\left(\boldsymbol{x}-\boldsymbol{x}_j,h\right)\varphi\left(\boldsymbol{x}_j\right) \tag{5.102}$$

Then we take the gradient to previous expression to provide the MLS gradient approximation as

$$\nabla\big(\hat{\varphi}(\boldsymbol{x})\big)=\nabla\hat{\varphi}(\boldsymbol{x})=\sum_j \nabla N\left(\boldsymbol{x}-\boldsymbol{x}_j,h\right)\varphi\left(\boldsymbol{x}_j\right) \tag{5.103}$$

Using the partition of nullity property, the previous approximation for the gradient can also be rewritten as

$$\nabla\hat{\varphi}(\boldsymbol{x})=\sum_j \nabla N\left(\boldsymbol{x}-\boldsymbol{x}_j,h\right)\left(\varphi\left(\boldsymbol{x}_j\right)-\varphi\left(\boldsymbol{x}\right)\right) \tag{5.104}$$

that evaluated in the particular point $\boldsymbol{x}=\boldsymbol{x}_i$ yields

$$\nabla\hat{\varphi}(\boldsymbol{x}_i)=\sum_j \nabla N\left(\boldsymbol{x}_i-\boldsymbol{x}_j,h\right)\left(\varphi\left(\boldsymbol{x}_j\right)-\varphi\left(\boldsymbol{x}_i\right)\right) \tag{5.105}$$

By comparing Eq. (5.90) and Eq. (5.105), it can be inferred that ∇N_{ji} and $\tilde{\psi}_j\left(\boldsymbol{x}_i\right)$ play the same role as approximations of the gradient of a scalar field. Eq. (5.90) read as

$$(\nabla\varphi)_i\approx\left(D_h\varphi\right)_i=\sum_j\left(\varphi_j-\varphi_i\right)\tilde{\psi}_j\left(\boldsymbol{x}_i\right) \tag{5.106}$$

And using Eq. (5.105), we can write

$$(\nabla\varphi)_i\approx\nabla\hat{\varphi}(\boldsymbol{x}_i)=\sum_j \nabla N\left(\boldsymbol{x}_i-\boldsymbol{x}_j,h\right)\left(\varphi\left(\boldsymbol{x}_j\right)-\varphi\left(\boldsymbol{x}_i\right)\right)=\sum_j \nabla N_{ji}\left(\varphi_j-\varphi_i\right) \tag{5.107}$$

However, we have not proved yet that $\tilde{\psi}_j\left(\boldsymbol{x}_i\right)=\nabla N_{ji}$. In the particular case of linear fields, the renormalized gradient derivative and the MLS approximations (with

at least a linear basis) reproduce exactly a linear field and therefore $(D_h\varphi)_i = \nabla\hat{\varphi}(\boldsymbol{x}_i)$ and we can conclude that ∇N_{ji} and $\tilde{\psi}_j^{\alpha}(x_i)$ for this particular field.

In order to ease the comparison with other formulations it is convenient to provide an alternative expression for $\tilde{\psi}_i(\boldsymbol{x}_j)$. Gizmo-MFV defines $\tilde{\psi}_i(\boldsymbol{x}_j)$ in Eq. (5.90) in terms of a renormalization matrix $B_H^{\alpha,\beta}$ satisfying condition given by Eq. (5.92). The matrix $B_H^{\alpha,\beta}$ is not the common renormalization matrix used in the SPH community. For instance, Lanson and Vila [Lanson & Vila, 2008a] define other renormalization matrix $(B_V^{\alpha,\beta})_i$ forcing the fulfillment of condition

$$\delta^{\alpha,\gamma} = \sum_{j\in} \left(B_V^{\alpha,\beta}\right)_i \frac{\partial W(\boldsymbol{x}_i - \boldsymbol{x}_j, h)}{\partial x_i^{\beta}} \left(x_j^{\gamma} - x_i^{\gamma}\right) V_j \tag{5.108}$$

The matrix used by Gizmo-MFV $B_H^{\alpha,\beta}$ is different from the matrix used in renormalized SPH-ALE $B_V^{\alpha,\beta}$. Even more, $B_V^{\alpha,\beta}$ is dimensionless but $B_H^{\alpha,\beta}$ has dimensions of $[L]^{-2}$. The renormalization matrix $B_H^{\alpha,\beta}$ has the computational advantage of operating over symmetric matrices.

Since the Kronecker function $\delta^{\alpha,\gamma}$ appears on both left hand side of Eq. (5.92) and Eq. (5.108), the following relation between matrices is obtained

$$\underbrace{\left(B_H^{\alpha,\beta}\right)_i \left(x_j^{\beta} - x_i^{\beta}\right) \psi_j(\boldsymbol{x}_i)}_{\tilde{\psi}_j^{\alpha}(\boldsymbol{x}_i)} = \left(B_V^{\alpha,\beta}\right)_i \frac{\partial W(\boldsymbol{x}_i - \boldsymbol{x}_j, h)}{\partial x_i^{\beta}} V_j \tag{5.109}$$

where the left hand side of Eq. (5.109) correspond with the definition of $\tilde{\psi}_j^{\alpha}(\boldsymbol{x}_i)$ in Eq. (5.91). Therefore we have obtained and alternative definition of $\tilde{\psi}_j^{\alpha}(\boldsymbol{x}_i)$ in terms of the kernel gradient $\nabla_i W_{ij}$

$$\tilde{\psi}_j^{\alpha}(\boldsymbol{x}_i) = \left(B_V^{\alpha,\beta}\right)_i \frac{\partial W(\boldsymbol{x}_i - \boldsymbol{x}_j, h)}{\partial x_i^{\beta}} V_j \qquad \tilde{\psi}_j(\boldsymbol{x}_i) = (B_V)_i \nabla_i W_{ij} V_j \tag{5.110}$$

Introduction of Eq. (5.110) into the semi-discrete form obtained by the Gizmo-MFV given by Eq. (5.98) gives

$$\frac{d(\boldsymbol{U}_i V_i)}{dt} + \sum_j (\mathbf{G}_{\boldsymbol{w}})_{ij} \cdot \left[V_i (B_V)_i \nabla_i W_{ij} V_j - V_j (B_V)_j \nabla_j W_{ji} V_i\right] = V_i \boldsymbol{S}_i \tag{5.111}$$

Omitting the renormalization process is equivalent to consider that renormalization matrix is the identity matrix, that is to say, $(B_V)_i = (B_V)_j = \mathbb{I}$. For that particular case the Gizmo-MFV formulation becomes the SPH-ALE formulation of Vila.

5.4.5. The Finite Point Method (FPM)

The Finite Point Method (FPM) is a meshless technique that was originally presented by Oñate et al. [Oñate et al., 1996b], [Oñate et al., 1996a] and [Oñate, 1998].

The solution of compressible Euler equations with FPM have been considered in detail in [Löhner et al., 2001], [Ortega et al., 2009] and [Ortega, 2014]. Bajko [Bajko et al., 2014], [Bajko, 2013] presented an accuracy improvement of the FPM that was applied for computational aeroacoustics.

FPM is based on the strong solution of PDE over a stationary cloud of points. It is a method that does not assign/calculate any weight to the points of the cloud (neither volume nor mass). These kinds of meshless methods resemble the FDM in the collocation approach and are usually designated as Generalized Finite Difference Methods (GFDM), because the grid layout can be considered a special case of a point cloud. Both the strong form and the stationary position of the points are important differences with other meshless formulations considered in the present section. The FPM approximates flow variables and their derivatives by means of a Weighted Least-Square procedure that is known as Fixed Least Squares (FLS). The MLSPH-ALE method uses an approximation based on Moving Least Squares, but the ALE capability allow us to set the frame velocity $\boldsymbol{w} = \boldsymbol{0}$, and with fixed point position MLS and FLS provide the same approximations. In the literature there are other GFDMs that share many features with the FPM, but differ in the technique used to approximate the partial derivatives on scattered data. Instead of using Least Squares with a polynomial basis, Jameson and Katz [Katz, 2009] have proposed the use of a Least Squares based on Taylor series.

In the same manner that FDM formulation can be derived from the perspective of FVM we propose to obtain the FPM formulation starting from the MLSPH-ALE method. The acronym FPM may be a bit confusing because it is also used for other meshless methods like the Finite Pointset Method [Kuhnert, 2003], [Suchde, 2018] and the Finite Particle Method [Liu et al., 2005] but we recall that in this work it is used for the Finite Point Method. In these two formulations particles can either be fixed in an Eulerian frame as interpolation points or can move in a Lagrangian frame.

We outline the derivation of the FPM following the works of [Löhner et al., 2001] and [Ortega et al., 2009].

We start the derivation from the Euler equations, written in differential conservative form:

$$\frac{\partial \boldsymbol{U}(\boldsymbol{x},t)}{\partial t} + \frac{\partial \boldsymbol{F}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}} = \boldsymbol{S}(\boldsymbol{x},t) \tag{5.112}$$

In order to obtain the semi-discrete form, the collocation method is adopted. The following expression should be satisfied in each point of the cloud

$$\left[\frac{\partial \hat{\boldsymbol{U}}(\boldsymbol{x},t)}{\partial t} + \frac{\partial \hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}}\right]_{\boldsymbol{x}=\boldsymbol{x}_i} = \left[\hat{\boldsymbol{S}}(\boldsymbol{x},t)\right]_{\boldsymbol{x}=\boldsymbol{x}_i} \tag{5.113}$$

where hats are used to denote the FLS approximations. We note that FLS shape functions do not interpolate point data, but assuming the approximation for the gradient

and source term Eq. (5.113) becomes

$$\frac{\partial \boldsymbol{U}(\boldsymbol{x}_i,t)}{\partial t} + \left[\frac{\partial \hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}}\right]_{\boldsymbol{x}=\boldsymbol{x}_i} = \boldsymbol{S}(\boldsymbol{x}_i,t) \tag{5.114}$$

In order to proceed further, Eq. (5.114) demands to evaluate the divergence of the flux. Noting that FLS corresponds with a particular case of MLS with fixed point position, the FLS approximation for the Eulerian flux is expressed in terms of $N(\boldsymbol{x}-\boldsymbol{x}_j)$ by

$$\hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t) = \sum_j N(\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{F}^{\alpha}_{\mathrm{E}}(\boldsymbol{x}_j,t) \tag{5.115}$$

By taking the divergence of the previous flux expression we obtain

$$\frac{\partial \hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}} = \sum_j \frac{\partial N(\boldsymbol{x}-\boldsymbol{x}_j)}{\partial x^{\alpha}}\boldsymbol{F}^{\alpha}_{\mathrm{E}}(\boldsymbol{x}_j,t) = \sum_j \nabla N(\boldsymbol{x}-\boldsymbol{x}_j)\mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_j,t) \tag{5.116}$$

At this moment the Partition of Nullity property $\sum_j \nabla N(\boldsymbol{x}-\boldsymbol{x}_j) = 0$ is invoked and a null term in the flux divergence is subtracted

$$\frac{\partial \hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}} = \sum_j \nabla N(\boldsymbol{x}-\boldsymbol{x}_j)\left(\mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_j,t) - \mathbf{F}_{\mathrm{E}}(\boldsymbol{x},t)\right) \tag{5.117}$$

We can now evaluate in the position of point i giving

$$\left[\frac{\partial \hat{\boldsymbol{F}}^{\alpha}_{\mathrm{E}}(\boldsymbol{x},t)}{\partial x^{\alpha}}\right]_{\boldsymbol{x}=\boldsymbol{x}_i} = \sum_j \nabla N(\boldsymbol{x}_i-\boldsymbol{x}_j)\left(\mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_j,t) - \mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_i,t)\right) \tag{5.118}$$

By using the compact nomenclature $\nabla N(\boldsymbol{x}_i-\boldsymbol{x}_j) \equiv \nabla N_j(\boldsymbol{x}_i) \equiv \nabla N_{ji}$ and inserting Eq. (5.118) into Eq. (5.114) the following semi-discrete expression is obtained

$$\frac{\partial \boldsymbol{U}(\boldsymbol{x}_i,t)}{\partial t} + \sum_j \nabla N_{ji}\left(\mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_j,t) - \mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_i,t)\right) = \boldsymbol{S}(\boldsymbol{x}_i,t) \tag{5.119}$$

This semi-discrete form is not stable. In order to stabilize the scheme, the following approximation is used

$$\begin{aligned} \mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_j,t) - \mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_i,t) &\approx 2\left(\mathbf{F}_{\mathrm{E}}\left(\frac{\boldsymbol{x}_i+\boldsymbol{x}_j}{2},t\right) - \mathbf{F}_{\mathrm{E}}(\boldsymbol{x}_i,t)\right) \\ (\mathbf{F}_{\mathrm{E}})_j - (\mathbf{F}_{\mathrm{E}})_i &\approx 2\big((\mathbf{F}_{\mathrm{E}})_{ij} - (\mathbf{F}_{\mathrm{E}})_i\big) \end{aligned} \tag{5.120}$$

Since $(\mathbf{F}_{\mathrm{E}})_{ij}$ is the flux in the midpoint of the segment connecting points i and j, it could be numerically evaluated with an approximate Riemann solver introducing the required dissipation , that is, $(\mathbf{F}_{\mathrm{E}})_{ij} \approx (\mathbf{G}_{\mathrm{E}})_{ij}$. Inserting the approximation given by Eq. (5.120) into Eq. (5.119) the final semi-discrete equation for the FPM read as

$$\frac{\partial \boldsymbol{U}_i}{\partial t} + 2\sum_j \nabla N_{ji}\big((\mathbf{G}_{\mathrm{E}})_{ij} - (\mathbf{F}_{\mathrm{E}})_i\big) = \boldsymbol{S}_i \tag{5.121}$$

where it can be observed that neither mass nor volume associated with the point appear in the semi-discrete form.

Obtaining FPM formulation from MLSPH-ALE

Starting with the MLSPH-ALE, the semi-discrete form for the Euler equations is given by Eq. (4.25), that we recall here for convenience

$$\frac{d(V_i \boldsymbol{U}_i)}{dt} + \sum_{j=1}^{n_i}\left[\frac{1}{2}\Big((\mathbf{F}_{\boldsymbol{w}})_j + (\mathbf{F}_{\boldsymbol{w}})_i\Big) - (\mathbf{F}_{\boldsymbol{w}})_i\right](-V_j\,\nabla N_{ij} + V_i\,\nabla N_{ji}) = V_i\,\boldsymbol{S}_i$$

For MLSPH-ALE method running in Eulerian mode the transport velocity $\boldsymbol{w} = \mathbf{0}$ and therefore $\mathbf{F}_{\boldsymbol{w}} = \mathbf{F}_{\mathrm{E}} - \boldsymbol{w} \otimes \boldsymbol{U} = \mathbf{F}_{\mathrm{E}}$. In Eulerian framework the time derivative $\dfrac{d\,(V_i \boldsymbol{U}_i)}{dt}$ is identical to $\dfrac{\partial\,(V_i \boldsymbol{U}_i)}{\partial t}$ and since particles volumes do not vary in time $\dfrac{\partial\,(V_i \boldsymbol{U}_i)}{\partial t} = V_i \dfrac{\partial \boldsymbol{U}_i}{\partial t}$. Taking into account these considerations the final semi-discrete system obtained with MLSPH-ALE method for the Euler equations in Eulerian framework read as

$$V_i \frac{\partial(\boldsymbol{U}_i)}{\partial t} + \sum_{j=1}^{n_i}\left[\frac{1}{2}\Big((\mathbf{F}_{\mathrm{E}})_j + (\mathbf{F}_{\mathrm{E}})_i\Big) - (\mathbf{F}_{\mathrm{E}})_i\right](-V_j\,\nabla N_{ij} + V_i\,\nabla N_{ji}) = V_i\,\boldsymbol{S}_i \tag{5.122}$$

By comparison of Eq. (5.121) and Eq. (5.122) it is inferred that we need to introduce particular settings in the MLSPH-ALE method that allow us to cancel the volume contributions. In order to factorize the volume V_i in the expression for the flux we can invoke several simplifications.

The simplest case considers that nodes are disposed in a uniform Cartesian grid. In that case $V_i = V_j$ and by symmetry $\nabla N_{ij} = -\nabla N_{ji}$. Inserting these relations in Eq. (5.122) the semi-discrete form becomes

$$\frac{\partial \boldsymbol{U}_i}{\partial t} + \sum_{j=1}^{n_i}\left[\frac{1}{2}\Big((\mathbf{F}_{\mathrm{E}})_j + (\mathbf{F}_{\mathrm{E}})_i\Big) - (\mathbf{F}_{\mathrm{E}})_i\right](2\nabla N_{ji}) = \boldsymbol{S}_i \tag{5.123}$$

that provides the same expression than that obtained for the FPM in Eq. (5.121) once the central flux $\frac{1}{2}\Big((\mathbf{F}_{\mathrm{E}})_j + (\mathbf{F}_{\mathrm{E}})_i\Big)$ is approximated by a numerical flux $(\mathbf{G}_{\mathrm{E}})_{ij}$. However, this particular case is not very interesting since a meshless formulation with points disposed in a regular grid does not provide any advantage over mesh-based methods.

We consider now a more general case with a fixed cloud of points with irregular distribution. In that scenario the volumes do not evolve in time and the volumes associated to particles may vary from one particle to another. Returning to Eq. (5.122) we focus the attention on the term inside brackets $(-V_j\,\nabla N_{ij} + V_i\,\nabla N_{ji})$. The gradients of the shape functions N_{ji} and N_{ji} can be expressed in terms of the kernel gradient, according to the approximations given in Eq. (5.82) and Eq. (5.83)

$$\nabla_i W_{ij}\,V_j \approx \nabla N_{ji} \qquad \nabla_j W_{ji}\,V_i \approx \nabla N_{ij} \tag{5.124}$$

Inserting the approximations given by Eq. (5.124) into Eq. (5.122) enables us to cancel the term V_i leading to

$$\frac{\partial(\boldsymbol{U}_i)}{\partial t} + \sum_{j=1}^{n_i}\left[\frac{1}{2}\Big((\mathbf{F}_{\mathrm{E}})_j + (\mathbf{F}_{\mathrm{E}})_i\Big) - (\mathbf{F}_{\mathrm{E}})_i\right](-V_j\,\nabla_j W_{ji} + \nabla_i W_{ji}\,V_j) = \boldsymbol{S}_i \tag{5.125}$$

since the kernel gradient is symmetric we can express the expression inside brackets of Eq. (5.125) as $2\nabla_i W_{ij}\,V_j$. Using the approximation $\nabla N_{ji} \approx \nabla_i W_{ij} V_j$ given in the left side of Eq. (5.124) we arrive at the same semi-discrete system given by Eq. (5.123), but for a general cloud of points that does not need to fulfill the conditions $V_i = V_j$ and symmetry $\nabla N_{ij} = -\nabla N_{ji}$.

5.5. Conclusions

In this chapter we have highlighted the existing connections between mesh-based and meshless methods for solving the Euler equations. The consideration of a third group that fills the gap between mesh-based and meshless methods can contribute to provide a coupling methodology between them. The MFVM method was used to show the connections with mesh-based methods and SPH formulations. The particle shape functions used by meshless-fv methods provide a versatile concept that can be related either with the cells/elements of mesh-based methods and with the kernel functions of SPH methods.

Meshless methods still have not achieved the development state of mesh-based methods. However, meshless methods are receiving great attention in different researching communities. These conditions gave rise to a myriad of meshless methods derived by different groups. Comparison between different formulations reveals that although procedures could vary in great extent the final semi-discrete equations have great similarities. In this section we have revisited the derivation of a set of meshless formulations and we have shown the examined methods can be obtained from particular settings of the MLSPH-ALE discretization. We exclude the boundary condition treatment in the comparison. However, in the derivation procedure we can clearly see that some methods ease the implementation of the boundary conditions with a boundary flux term. We have also shown that MLSPH-ALE is a general formulation that includes many of the examined methods.

The MLSPH-ALE method enjoys the advantages of a very flexible selection of particle shape functions. The FVPM was derived by exploding the derivation properties of the simplest shape function satisfying the partition of unity property. The shape functions used by the FVPM correspond with the MLS shape functions using a polynomial basis with a constant term. Moreover, The FVPM is the procedure that gives the clearest interpretation of the interaction flux in the overlap region of particles. In the MLSPH-ALE method the procedure to define this effective flux is not unique.

On the other hand, the MFVM provides a semi-discrete form where the interaction flux is not fully discretized in the particle positions. By assuming a constant interaction flux in the MFVM we obtain the same continuous effective area that the one obtained in MLSPH-ALE method.

SPH-ALE and Gizmo-MFV obtain a discrete effective area in terms of the gradient of the kernel and the volume associated to particles. In the MLSPH-ALE method the effective area is a symmetric expression given in terms of the gradient of the particle shape functions and the volume of the particles. In general, we can not provide an analytical expression that relates the kernel gradient with the gradient of a particle shape function obtained with MLS. MLS shape functions are calculated by means of a weighted least squares approach. When the weighted function used by MLS is exactly the kernel function we can obtain an analytical expression relating the kernel gradient with the shape function gradient. By comparison of the kernel and MLS approximations of the gradient of a function we manage to make some term identifications to prove that MLSPH-ALE can be particularized to arrive at the semi-discrete equations derived with SPH-ALE and Gizmo-MFV methods.

The last part of the chapter was devoted to trace a path relating the MLSPH-ALE method with the Finite Point Method. This is a very interesting fact because it is proved that the Eulerian version of the formulation contains a fixed meshless method formulation that have been extensively tested in subsonic and transonic flow.

Chapter 6

The MLSPH-ALE method as a candidate for implicit LES

6.1. Introduction

Turbulence is very computationally demanding due to the wide range of spatial and temporal scales involved. Turbulence is such a complex phenomenon that even in the literature there is not available a precise definition. However, turbulent flows are ubiquitous in nature and there is an agreement in the following features: Apparently random and chaotic behavior; dependence on initial conditions; a wide range of length and time scales; three-dimensional, time-dependent and rotational character represented by eddies; time and space intermittency; increase diffusion and dissipation rates [Pope, 2000; Tennekes, 1972; Kundu et al., 2015].

The intention here is to present the common approaches to compute turbulent flows in meshless methods. The different approaches to deal with turbulence flows are based on the selection of a subrange of scales to be captured and another one to be modeled. Representative scales of the eddies and the energy cascade concept introduced by Richardson and Kolmogorov are depicted in Figure (6.1). The smallest scale of the eddies associated with the dissipation process is denoted with ν meanwhile the eddies in the largest size range are characterized by the length scale ℓ_0 which is comparable to the flow scale $\mathcal{L}$.

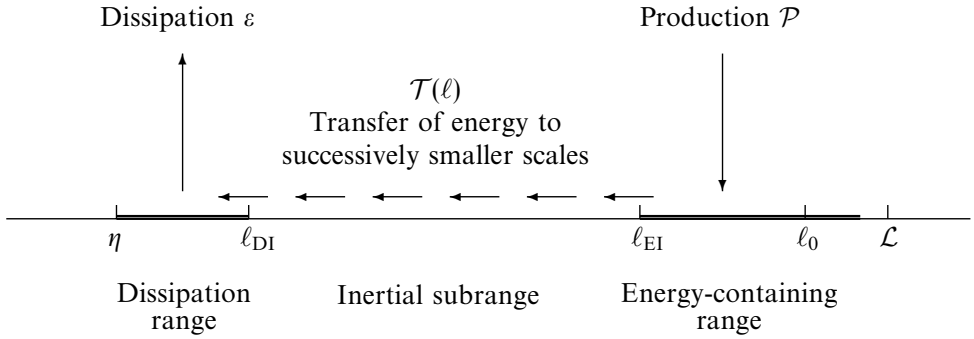


Figure 6.1. Energy cascade with the characteristic length scales for turbulent flows. Adapted from [Pope, 2000]

In the same manner that in mesh-based methods the most common approaches to compute turbulent flows with meshless methods are: Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes(RANS). However, meshless methods were predominantly expressed in a Lagrangian framework and therefore some turbulence models for meshless methods differ from their counterparts in mesh-based methods. Following, an outline of the development of turbulence models in meshless methods is provided.

Monaghan conducted the first attempt to introduce a turbulence model in a SPH formulation. The proposed formulation was labeled as α-SPH [Monaghan, 2002] and is based on the Lagrangian-Averaged Navier-Stokes turbulence model proposed by Holm [Holm, 1999]. In a later publication Monaghan [Monaghan, 2012] adopt the same procedure but using a more efficient smoothing algorithm for the velocity. The enhanced model was called as SPH-ϵ model. Assuming a weakly compressible approach and using the SPH-ϵ model Robinson and Monaghan [Robinson & Monaghan, 2011] performed computations of decaying turbulence in a two-dimensional no-slip wall-bounded domain. Adami et al. [Adami et al., 2012b] proposed a turbulence model that is based on the transport velocity concept introduced by [Monaghan, 2002] with the novelty of incorporating of a constant background pressure that regularizes the particle motion. Authors test the model with three-dimensional Taylor-Green vortex flow and achieve satisfactory dissipation rates in the range of Re from 100 to 3000 when compared with DNS data of Brachet [Brachet et al., 1983].

A great number of publications dealing with compressible turbulence in meshless methods were developed with astrophysical applications in mind. Bauer and Springel [Bauer & Springel, 2012] compared the accuracy of SPH and Moving-Mesh simulation codes dealing with subsonic, transonic and supersonic turbulence. Authors reported that traditional SPH formulations yields problematic results in the subsonic regime. There is some controversy in the causes of failure. According to Price [Price, 2011], the inability of SPH methods to capture a Kolmogorov-like spectrum in subsonic regime is caused by the excessive artificial viscosity. Price has shown that the use of viscosity

switches can remedy some issues of traditional SPH methods with subsonic turbulence. Hopkins [Hopkins, 2015] confirmed the limitations of traditional SPH methods in subsonic turbulence and also reported that Meshless Finite Volume Methods (MFVM) using Riemann solvers offer a significant improvement. Hopkins also detected that MFVM methods lose accuracy relative to stationary grid codes for highly subsonic problems once the Mach numbers fall in the range of ≈ 0.01. Exploding the advantage that MFVM can be run in Eulerian mode Hopkins demonstrated that the loss of accuracy in MFVM in highly subsonic problems is caused by the Lagrangian movement of particles.

Another front of development of turbulence models in meshless methods is pushed by researchers dealing with open channel flows and wave hydrodynamics in coastal and oceanic flows. The two approaches predominantly used by researching in these fields to compute turbulent flows are RANS models and sub-particle scale (SPS) models embedded in LES framework. The first publication that implements a $k-\epsilon$ turbulence RANS model into a particle method was conducted by Violeau et al. [Violeau et al., 2002]. Shao et al. [Shao, 2006] implemented the $k-\epsilon$ in an incompressible SPH method to simulate wave breaking and overtopping on a sloping sea wall. Gotoh et al. [Gotoh et al., 2001] implemented the first LES turbulence model in a particle method. Shao and Gotoh [Shao & Gotoh, 2005] adapted the particle LES model to an incompressible SPH formulation meanwhile Dalrymple et al. [Dalrymple & Rogers, 2006] conducted the extension of particle LES in a weakly compressible SPH formulation. Di Mascio et al. [Di Mascio et al., 2017] rewrote the δ-SPH method to accommodate the LES turbulence model resulting in the δ-LES-SPH scheme. Parameters associated with stabilization dissipative terms are computed with a Smagorinsky subgrid scale model. Krimi et al. [Krimi et al., 2020] proposed to use the Automatic Dissipation Method (ADA) [Li & Tsubokura, 2017] in the family of δ-SPH methods. The results show that the proposed numerical method is able to simulate complex flows. It alleviates the parameter dependency of δ-SPH methods and it is generally less dissipative than the δ-LES-SPH model.

In this chapter we will test the ability of the MLSPH method to perform implicit Large Eddy Simulations (iLES) of isotropic turbulence. This is a first step towards the final goal of developing a very accurate meshless method for the computation of turbulent flows.

6.2. Application of MLSPH-ALE to isotropic turbulence

6.2.1. Decay of compressible isotropic turbulence

In this numerical example, we extend the proposed formulation to 3D to analyze the decay of compressible isotropic turbulence. This test case is commonly used to verify the ability of the numerical method to simulate turbulent flow. Many authors have used this numerical test to develop new subgrid scale models and to analyze numerical

methods [Lee et al., 1991; Rizzetta et al., 1999; Spyropoulos & Blaisdell, 1996; Hickel et al., 2014; Kotov et al., 2016].

Setup of the problem

A periodic $2\pi \times 2\pi \times 2\pi$ cube is considered for the computational domain and it is discretized using an homogeneous distribution of particles. Periodic boundary conditions are considered for all the boundaries. As it is usual in this problem, the flow is defined by the turbulent Mach number (M_t) and Taylor's micro-scale Reynolds number (Re_λ)

$$M_t = \frac{\sqrt{\langle u_i' u_i' \rangle}}{\langle c \rangle} \tag{6.1}$$

$$Re_\lambda = \frac{\langle \rho \rangle \, u_{rms}' \lambda}{\mu} \tag{6.2}$$

where the notation $\langle\rangle$ refers to mean value and primes denote fluctuating variables. The root mean square of the velocity (u_{rms}) is defined as

$$u_{rms} = \sqrt{\frac{\langle u_i' u_i' \rangle}{3}} \tag{6.3}$$

The initial conditions are defined following a given initial three-dimensional kinetic energy spectrum as

$$E_{3D} \sim k^4 \mathrm{e}xp \left[-2 \left(\frac{k}{k_0} \right)^2 \right] \tag{6.4}$$

where k is the magnitude of the wave number vector, and $k_0 = 4$ is the wavenumber at the peak of the spectrum.

In this case, the initial velocity fluctuations are parametrized by the turbulent Mach number and also by the fraction of energy in the dilatational part of the velocity, $\chi = 0.2$ [Sarkar et al., 1991]. The initial turbulent Mach number is taken as $M_{t,0} = 0.4$ and the initial Taylor's microscale Reynolds number is chosen as $Re_{\lambda,0} = 2157$, which corresponds with a Reynolds number $Re = 536.9$. The setup presented here corresponds to the case 6 of [Spyropoulos & Blaisdell, 1996].

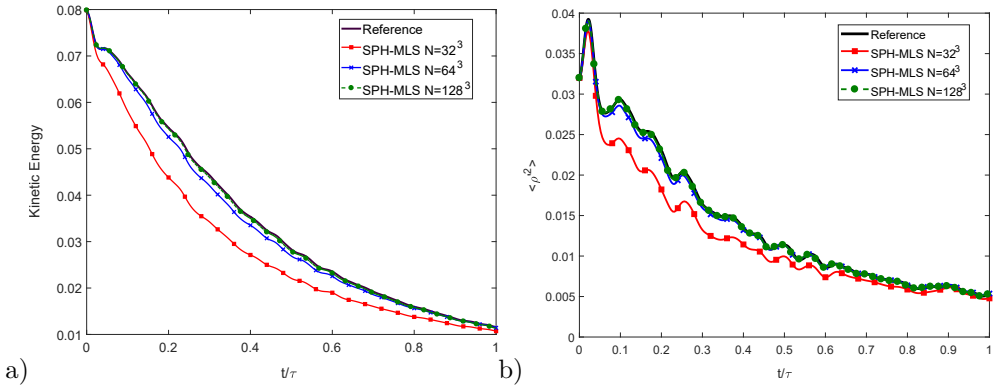
With this configuration, the initial density and temperature fields are given by

$$\begin{aligned} \left(\rho_{rms}'\right)^2 / \langle \rho \rangle^2 &= 0.032 \\ \left(T_{rms}'\right)^2 / \langle T \rangle^2 &= 0.005 \end{aligned} \tag{6.5}$$

The evolution of the flow with this initial conditions leads to a nonlinear subsonic regime [Garnier et al., 2009] and weak shocklets develop spontaneously from the turbulent motion. Three different particle discretizations are used in this test case: 32^3, 64^3 and 128^3 particles.

We run this test case using a time step of $\Delta t = 0.05$ with the 32^3 particle distribution. This corresponds to 250 time-steps per eddy turnover time τ, that is $\tau = 12.5$ for this configuration. For the 64^3 and 128^3 particle distributions we keep the CFL constant. In figure 6.2 we show the results obtained with the present approach in terms of the time evolution of the kinetic energy and the mean square density fluctuations. The results are compared with a reference solution computed with a sixth-order compact finite difference scheme, with explicit filtering using a tenth-order Padé filter [Visbal & Rizzetta, 2002]. This solution follows closely the results of a DNS.

The decay of kinetic energy is plotted in Figure 6.2 a). The coarser discretization shows excessive dissipation. However, it can be seen that as the number of particles is increased, the results converge to the reference solution, and the results obtained for 128^3 particles are in excellent agreement with those of the reference solution. The results obtained for the density fluctuations follow the same trend, as shown in Figure 6.2 b). Figure 6.3 a) shows that the proposed scheme is able to reproduce the thermodynamic variables of the flow. This is a remarkable result, since some methods are not able to simultaneously predict the correct scaling and decay rates of thermodynamic variables [Honein & Moin, 2004].



a) b)

Figure 6.2. Decay of homogeneous isotropic turbulence. Time evolution of kinetic energy (left) and mean square density fluctuations (right). The reference solution was computed following [Visbal & Rizzetta, 2002].

The instantaneous three-dimensional energy spectra ($E(k) = \rho((u^x)^2 + (u^y)^2 + (u^z)^2)$) at $t/\tau = 0.3$ is plotted in Figure 6.3 b). The results reproduce correctly the reference spectrum, although some overdissipation is found for the coarser grid. It is also observed that the proposed scheme is able to reproduce the two different slopes appearing in the energy spectrum, which agrees with the Eddy-Damped Quasi-Normal Markovian Theory (EDQNM) [Hussaini, 1998]. As the number of particles is increased, the spectrum converges to the reference. It is important to note that no piling-up of energy is detected.

In Figure 6.4 the Q-criterion iso-surface with a value of 0.25 is compared at $t = 10$ for the different particle resolutions. It can be noted the effect of the particle resolution.

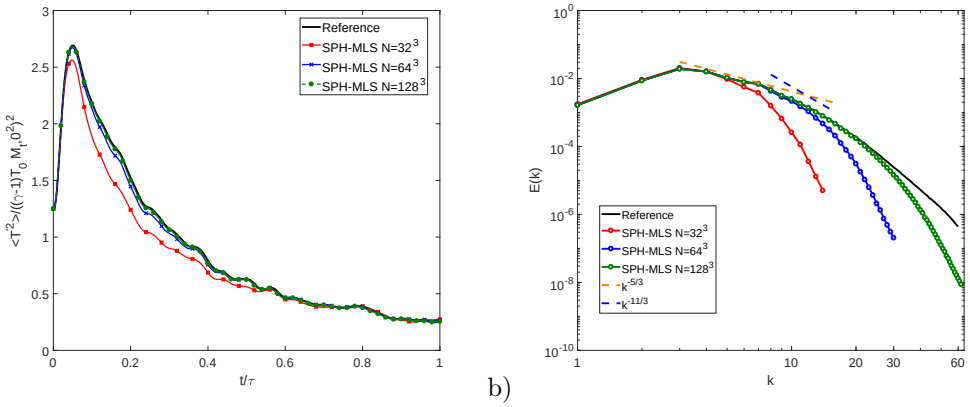


Figure 6.3. Decay of homogeneous isotropic turbulence. Time evolution of normalized temperature fluctuations (left) and instantaneous three-dimensional energy spectra at $t/\tau = 0.3$ (right). The reference solution was computed following [Visbal & Rizzetta, 2002].

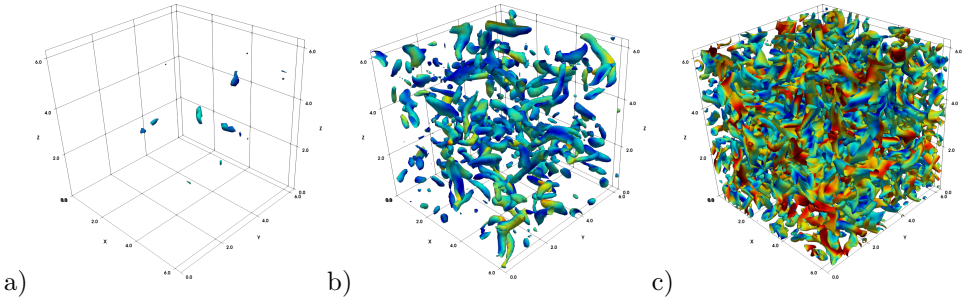


Figure 6.4. Decay of homogeneous isotropic turbulence. Iso-surfaces of Q-Criterion$= 0.25$ at $t = 10$, where a) is obtained with 32^3 particles, b) with 64^3 particles and c) with 128^3 particles.

6.2.2. 3D Taylor–Green Vortex

The 3D Taylor–Green vortex (TGV) test case is commonly used to test the accuracy and performance of numerical methods on the direct numerical simulation of a three-dimensional periodic and transitional flow. The computational domain is the cube defined as $[0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$ with periodic boundary conditions.

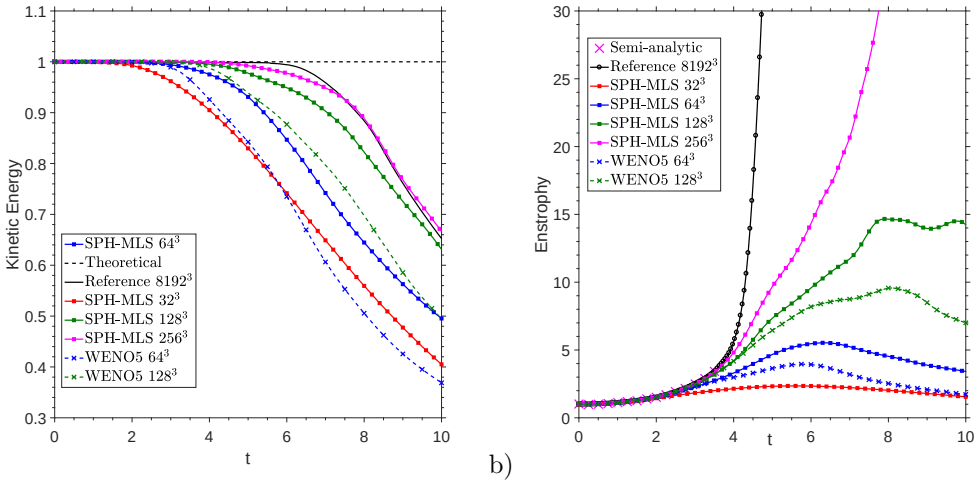
Two different test cases are analyzed here. The inviscid TGV and a viscous TGV with a Reynolds number $Re = 1600$. These test cases are solved using three different particle distributions with 32^3, 64^3, 128^3 and 256^3 particles, until a final time of $t = 10$.

TGV: Inviscid test case

The initial condition of the inviscid TGV is defined as

$$\begin{aligned} u^x(x,y,z,0) &= \sin(x)\cos(y)\cos(z) \\ u^y(x,y,z,0) &= -\cos(x)\sin(y)\cos(z) \\ u^z(x,y,z,0) &= 0 \\ \rho(x,y,z,0) &= 1 \\ p(x,y,z,0) &= 100 + \frac{1}{16}[(\cos(2x)+\cos(2y))(2+\cos(2z)) - 2] \end{aligned} \tag{6.6}$$

In this test case, the ratio of specific heat is $\gamma = 5/3$. The inviscid case is equivalent to an infinite Reynolds number. Ideally, in this test case, there is no decay of the kinetic energy. However, since the numerical method introduces some dissipation, all the dissipation comes from the numerical scheme. When the flow scales are smaller enough, the solution becomes under-resolved, and dissipation (and also dispersion) errors are introduced by the numerical scheme. The numerical dissipation introduced by the numerical scheme starts a decay process which ideally should follow the (-5/3) decay Kolmogorov's law. The evolution of the kinetic energy and enstrophy obtained by the proposed scheme are plotted in Figure 6.5 a), and compared with the results obtained by the WENO5 scheme on a FD framework [Shu et al., 2005], with a reference solution obtained by [Brachet et al., 1983] with a very fine grid (8192^3) and the semi-analytic results of [Fehn et al., 2022].



a) b)

Figure 6.5. 3D inviscid Taylor–Green: Time evolution of the kinetic energy a) and the enstrophy b).

It can be seen that the proposed meshless method converges to the Reference solution as the number of particles is increased. Moreover, it obtains more accurate results

than the WENO5 scheme. An important result is that the slope of the kinetic energy decay is similar to that of the reference solution. The enstrophy levels are also larger than those obtained by the WENO5 scheme, indicating the presence of more energetic vortices in the solution obtained by the SPH-MLS scheme.

TGV: Viscous case with $Re = 1600$

Finally, we present the results for the viscous TGV. The initial condition for this test case is a slight variation of that of the inviscid case to match the case description provided in [Garmann & Visbal, 2015].

$$\begin{aligned} u^x(x,y,z,0) &= \sin(x)\cos(y)\cos(z) \\ u^y(x,y,z,0) &= -\cos(x)\sin(y)\cos(z) \\ u^z(x,y,z,0) &= 0 \\ p(x,y,z,0) &= \frac{1}{\gamma M_0^2} + \frac{1}{16}\left[(\cos(2x)+\cos(2y))(2+\cos(2z))-2\right] \\ \rho(x,y,z,0) &= \gamma M_0^2 p(x,y,z,0) \end{aligned} \tag{6.7}$$

A uniform dimensionless temperature field of unity has been assumed along with the perfect gas relation. The reference Mach number is chosen as $M_0 = 0.1$ to minimize compressibility effects. A constant Prandtl number of $Pr = 0.71$, $\gamma = 1.4$ and a Reynolds number of $Re = 1600$ are used. With this setup, the flow evolves and smaller scales progressively appear, until the flow transitions to turbulence and decays in a similar way to decaying homogeneous turbulence.

The evolution of the kinetic energy and enstrophy obtained by the proposed scheme are plotted in Figure 6.6, and compared with the results obtained by the WENO5 scheme [Shu et al., 2005] and the DNS solution presented in [Garmann & Visbal, 2015].

As in the previous case, the solution converges to the DNS solution as the particle distribution is increased, and it obtains consistently more accurate results than the WENO5 scheme for both, the decay of kinetic energy and the enstrophy evolution. In Figure 6.7 we show the time evolution of the Q-Criterion from $t = 0$ to $t = 10$ with the finer particle resolution. It can be observed the flow transition from laminar with vortex-tubes into small-scale turbulence.

6.3. Conclusions

In this chapter MLSPH-ALE method was tested to compute turbulent flows. An asset of this formulation is the ability to reproduce the Kolmogorov cascade in homogeneous turbulence with an under-resolved discretization.

MLSPH-ALE method was able to capture the energy spectrum in decaying turbulent compressible flow. As the resolution is increased, the energy spectra approaches to

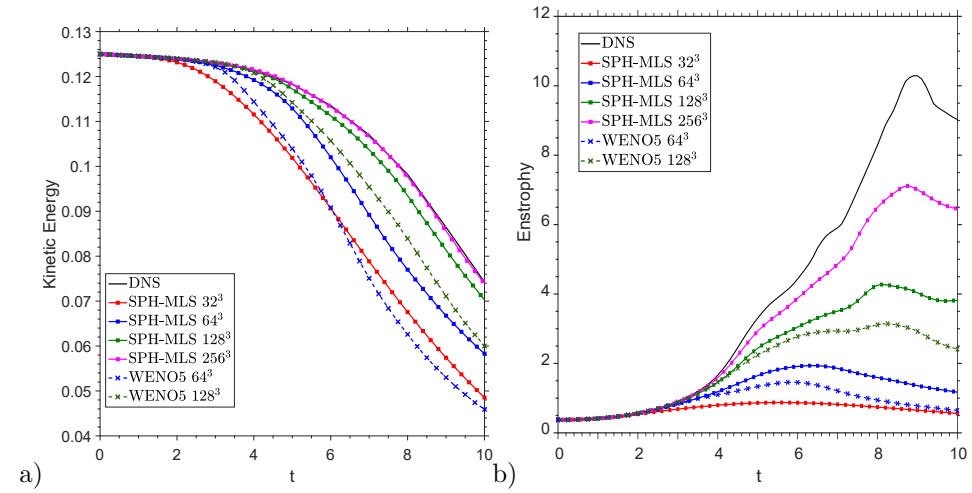


Figure 6.6. 3D viscous Taylor–Green: Time evolution of the kinetic energy a) and the enstrophy b).

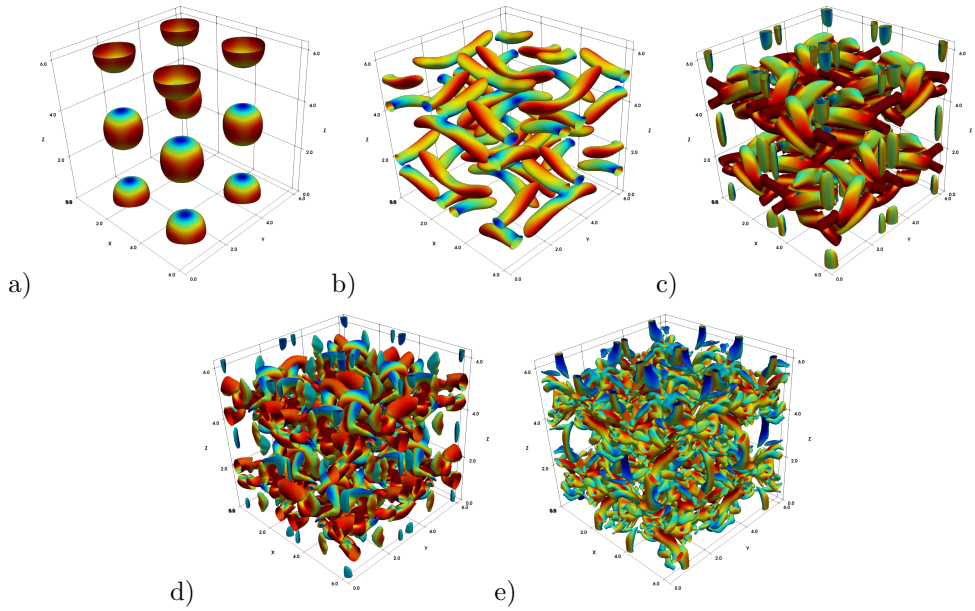


Figure 6.7. 3D viscous Taylor–Green: Iso-surfaces of Q-Criterion with a value of 0.5**: a) at** $t = 0$**, b) at** $t = 2.5$**, c) at** $t = 5$**, d) at** $t = 7.5$ **and e) at** $t = 10$ **with** 128^3 **particles.**

the reference solution obtained with DNS. Additionally, the model predict the reference decay of the thermodynamic variables involved.

Simulations of the three-dimensional Taylor—Green vortex probe that MLSPH-ALE method can capture with an adequate resolution the transition from laminar to turbulent flow. Results show that the temporal evolution of kinetic energy and

dissipation rates converge to DNS profiles when the spatial resolution is increased.

The results obtained in the turbulent cases considered are promising. However, further steps need to be given to implement a high accuracy meshless method to compute general turbulent flows. Since the model has only been tested in a Cartesian layout in Eulerian framework the effect of the irregular placement and movement of particles was not studied. Another desirable extension is to consider near-wall turbulent flows. Finally, the use of a method for adaptive dissipation in an iLES framework could improve the accuracy of the results.

Chapter 7

Conclusions

7.1. Conclusions

This thesis has developed and analyzed a generalized meshless formulation that overcomes some of the drawbacks of current CFD state-of-the-art methods. The meshless formulation adopts an ALE framework, it is based on Riemann solvers, uses MLS as approximation technique over scattered data and includes a boundary term that eases the introduction of boundary conditions. Moreover, the accuracy of the proposed method is comparable to that of standard grid-based methods.

- In Chapter 3, a high-accurate meshless formulation for solving the incompressible Navier-Stokes equations with the weakly compressible approach is developed. The formulation presented in this chapter is an enhanced formulation of the SPH-ALE scheme proposed by Vila. The novelties of the formulation rely on use of the MLS-based technique for two different purposes. The first task of MLS reconstruction is to conduct a high order reconstruction of the Riemann states at the interface between interacting particles. The second task of MLS is to estimate the viscous stress in the interface between interacting particles. In order to provide stabilization, the *a-posteriori* MOOD procedure is adopted. For compressible liquid flows with the presence of discontinuities the presented method improves the results obtained with other SPH-ALE methods using a MUSCL strategy with flux limiters. Underwater explosions [Liu et al., 2003b], cavitating flows [Pineda et al., 2019] and laser-induced shocks for liquid metal droplets [Koukouvinis et al., 2018] are some examples of applications that can take advantage of the improvements introduced in the proposed formulation.

- In Chapter 4, a high-accurate meshless formulation (MLSPH-ALE) was derived for solving Navier-Stokes equations.

The method was derived starting from the differential equation of a conservation law expressed in ALE form. This new formulation uses MLS as approximation technique circumventing consistency deficiencies of kernel approximation. The derivation of the semi-discrete form invokes properties of MLS functions that ease the integration of the weak form and assure the local conservativeness. The new method provides a boundary term that enable imposing wall boundary conditions without requiring ghost particles. The results obtained with this method are in close agreement with the results obtained in the bibliography with mesh-based methods.

- In Chapter 5, the MLSPH-ALE method is stated as a generalized meshless finite volume formulation.
  Under some particular settings the new method yields the same semi-discrete equations that other meshless methods published in the literature. The SPH-ALE method proposed by Vila [Vila, 1999] and the FVPM proposed by Hietel et al. [Hietel et al., 2000] are renown meshless formulations that can be derived starting from the MLSPH-ALE method.

- In Chapter 6, the MLSPH-ALE method is applied to solve turbulent flow problems.
  It is shown that the use or Riemann solvers in the formulation of the MLSPH-ALE method allows to solve problems in an iLES framework. The decay of compressible isotropic turbulence and the Taylor–Green vortex test cases are used to test the ability of the method to deal with turbulent and transitional flows.

7.2. Future research lines

Although the high-accurate meshless framework proposed in this thesis solves some of the problems of current meshless formulations, there are still open paths to continue the research: add extensions, improve the performance of the algorithm, application the formulation to a particular industrial problem. Some of the possibilities to extend this research activity in the future are summarized below:

- Implement a projection technique to solve incompressible flows.
  In Chapter 3 we solve incompressible flows by adopting a weakly compressible approach. Weakly compressible codes are usually tested against analytical solutions of incompressible flow such as Poiseuille or Taylor–Green flows. A projection scheme avoids the time step restriction that must fulfill explicit formulations. Moreover, incompressible flows avoid the use of preconditioners and fixes that require compressible formulations when they are applied in low Mach flows. On the other hand, they are not able to deal with discontinuities and the implementation is more complex.

- As expected, the Eulerian scheme is faster than the Lagrangian method. Then, a possible way for improving the efficiency of the proposed method is to combine Eulerian and Lagrangian particles. This idea has been explored previously in the context of ISPH [Fourtakas et al., 2018] and fits very naturally in the proposed formulation.

- Implementation of a local Voronoi tessellation.
  MLSPH-ALE method has the same stencil for calculating the MLS shape functions that for evaluating the fluxes in the sum over neighbors. A local Voronoi tessellation for each particle could be very useful to calculate the flux over a reduced number of neighbors and could also enable to define a more precise stencil for implementing the MOOD stabilization technique.

- Treatment of shocks in presence of walls.
  In Chapter 3 the MOOD procedure was applied for shock wave propagation in liquids in 1D and 2D configurations without boundaries. In Chapter 4 the wall boundary condition was applied on the walls of a 2D cylinder in an ideal flow. However, we have not considered any test problem where a shock wave interacts with a wall, in complex geometries.

- Coupling of FVM-MLS and MLSPH-ALE.
  In this thesis the MLSPH-ALE method is shown to have a close similarity with FVM. This feature could be explored to implement a coupling strategy between the meshless MLSPH-ALE and the FVM-MLS. This could be advantageous in boundary layer flows.

- Free Surface Flows and Fluid-Structure Interaction
  There is a myriad of applications that demands further extensions of meshless methods to deal with Free Surface Flows (FSF) and Fluid-Structure Interaction (FSI) phenomena. For instance, nowadays Offshore Wind Power is a very competitive sector that demands efficient designs to cope with FSI phenomena induced by violent FSF. The damages caused by natural disasters like floods, tsunamis and lava flows could be reduced if high risk zones could be improved by conducting simulations anticipating the catastrophic events. In the current state-of-the-art it is noticed that most open-source projects developing meshless methods are more focused on improving the computational aspects of the algorithm than in improving the numerical method itself. Some interesting problems considered in the literature to test FSF and FSI is the dam-break flow through an elastic gate [Antoci et al., 2007; Li, 2013] and the dam-break on elastic wall [Walhorn et al., 2005].

Appendix A

The Cubic Spline Kernel

A.1. Introduction

The idea behind smoothed particle hydrodynamics is to imagine that each computational particle j of mass m_j in position $\boldsymbol{x}_j$ represents a distribution of density and not a localized mass point. The contribution to the density at point $\boldsymbol{x}$ arising from a particle j at $\boldsymbol{x}_j$ can be expressed by

$$\rho_j(\boldsymbol{x}) = m_j W\left(\left|\boldsymbol{x} - \boldsymbol{x}_j\right|, h\right)$$

where $W\left(\left|\boldsymbol{x} - \boldsymbol{x}_j\right|; h\right)$ is the smoothing kernel that describes the form of the density distribution of the computational particle centered in $\boldsymbol{x}_j$. When thinking about smoothed particle hydrodynamics, it helps to use the example of a Gaussian (spherically symmetric) kernel.

$$W\left(\left|\boldsymbol{x} - \boldsymbol{x}_j\right|; h\right) = \frac{\sigma}{h^d} \exp\left[-\left(\left|\boldsymbol{x} - \boldsymbol{x}_j\right| / h\right)^2\right]$$

where d refers to the number of spatial dimensions and σ is a normalization factor given by $\sigma = \left[\frac{1}{\sqrt{\pi}}, \frac{1}{\pi}, \frac{1}{\pi\sqrt{\pi}}\right]$ in $[1, 2, 3]$ dimensions. By comparison with the probability density function of a Gaussian distribution it can be noticed that $\boldsymbol{x}_j$ acts as the mean and the smoothing length h is related to the standard deviation σ of the distribution by the expression $h = \sqrt{2}\sigma$.

The physical density at a given point is obtained by summing the overlapping contribution from all of the particles j.

$$\rho(\boldsymbol{x}) = \sum_{j=1}^{N} \rho_j(\boldsymbol{x}) = \sum_{j=1}^{N} m_j W\left(\left|\boldsymbol{x} - \boldsymbol{x}_j\right|; h\right)$$

Next, details about the determination of the normalization constants required by the cubic spline kernel are provided.

A.1.1. Cubic Spline Kernel

The cubic spline kernel is given by

$$W(q,h) = \frac{\alpha_d}{h^d} f(q) = \frac{\alpha_d}{h^d} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & q \leqslant 1 \\ \frac{1}{4}(2-q)^3, & 1 < q \leqslant 2 \\ 0, & q > 2 \end{cases}$$

with $q = \frac{|\boldsymbol{x}-\boldsymbol{x}'|}{h}$ the normalized distance and h the smoothing length. Its derivative with respect to the normalized distance reads as

$$\frac{\partial W(q,h)}{\partial q} = \frac{\alpha_d}{h^d} \begin{cases} -3q + \frac{9}{4}q^2 & q \leqslant 1 \\ -\frac{3}{4}(2-q)^2 & 1 < q \leqslant 2 \\ 0 & q > 2 \end{cases}$$

Normalization constant in a 1-D Domain

$0 \leq q \leq 1$:

$$\int_0^h \left(1 - \frac{3}{2}q^2 + \frac{3}{4}q^3\right) dx = \frac{11h}{16}$$

$1 < q \leq 2$:

$$\int_h^{2h} \frac{1}{4}(2-q)^3 dx = \frac{1h}{16}$$

In the whole support of the kernel (left and right):

$$\int_\Omega W d\Omega = \frac{\alpha_1}{h}\left[2\left(\frac{11h}{16} + \frac{1h}{16}\right)\right] = 1$$

resulting

$$\alpha_1 = \frac{2}{3}$$

Normalization constant in a 2-D Domain

$0 \leq q \leq 1$:

$$\int_0^{2\pi}\int_0^h \left(1 - \frac{3}{2}q^2 + \frac{3}{4}q^3\right) r dr d\theta = \frac{11\pi h^2}{20}$$

$1 < q \leq 2$:

$$\int_0^{2\pi}\int_h^{2h} \frac{1}{4}(2-q)^3 r dr d\theta = \frac{3\pi h^2}{20}$$

In the whole support of the kernel:

$$\int_{\Omega} W d\Omega = \frac{\alpha_2}{h^2}\left(\frac{11\pi h^2}{20} + \frac{3\pi h^2}{20}\right) = 1$$

resulting

$$\alpha_2 = \frac{10}{7\pi}$$

Normalization constants in a 3-D Domain

Integrating in the first octant:
$0 \leq q \leq 1$:

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \int_0^{h} \left(1 - \frac{3}{2}q^2 + \frac{3}{4}q^3\right) r^2 \sin\theta dr d\theta d\phi = \frac{19\pi h^3}{240}$$

$1 < q \leq 2$:

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \int_h^{2h} \frac{1}{4}(2-q)^3 r^2 \sin\theta dr d\theta d\phi = \frac{11\pi h^3}{240}$$

In the whole domain support of the kernel (8 octants):

$$\int_{\Omega} W d\Omega = \frac{\alpha_3}{h^3}\left[8\left(\frac{19\pi h^3}{240} + \frac{11\pi h^3}{240}\right)\right] = 1$$

resulting

$$\alpha_3 = \frac{1}{\pi}$$

Appendix B

Time derivative of an integral

B.1. Introduction

B.1.1. Reynolds Transport Theorem (RTT)

Let f be any scalar, vector or tensor function and $\widehat{\Omega}(t)$ a material volume in time t. We denote with $F(t)$ the total amount of f that carries the material volume $\widehat{\Omega}(t)$

$$F(t) = \int_{\widehat{\Omega}(t)} f dx^d \tag{B.1}$$

In Continuum Mechanics a material volume is a system that contains the same particles. The governing equations in Continuum Mechanics are obtained by application of physical laws to material volumes and thus is very important to analyze the temporal derivative $F(t)$ given by

$$\frac{D}{Dt}F(t) = \frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}} \int_{\widehat{\Omega}(t)} f dx^d \tag{B.2}$$

In the two first expressions we use the convention of denoting material time derivative with capital letters and on the last we use a partial time derivative keeping fixed the material coordinates $\boldsymbol{X}$.

According to the definition of the derivative

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}} \int_{\widehat{\Omega}(t)} f dx^d = \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\int_{\widehat{\Omega}_{t+\Delta t}} f(\boldsymbol{x}, t+\Delta t) dx^d - \int_{\widehat{\Omega}_t} f(\boldsymbol{x}, t) dx^d\right) \tag{B.3}$$

where $\widehat{\Omega}(t)$ is the material volume in time t and $\widehat{\Omega}_t$ denotes the spatial domain occupied by the material system at time t and $\widehat{\Omega}_{t+\Delta t}$ the spatial domain occupied by the same material system at time $t + \Delta t$. By transforming both integrals of the RHS to the material domain in time t_0

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \lim_{\Delta t\to 0}\frac{1}{\Delta t}\left(\int_{\widehat{\Omega}_0} f(\boldsymbol{X},t+\Delta t)J(\boldsymbol{X},t+\Delta t)dX^d - \int_{\widehat{\Omega}_0} f(\boldsymbol{X},t)J(\boldsymbol{X},t)dX^d\right) \tag{B.4}$$

With this change in the domain of integration, f becomes a function of the material coordinates and the domain of integration is independent of time

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_0}\left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}} \left(f(\boldsymbol{X},t)J(\boldsymbol{X},t)\right) dX^d \tag{B.5}$$

Using the product rule for derivatives and expression for temporal derivative of the determinant of the Jacobian $\frac{DJ}{Dt} = J\frac{\partial u_i}{\partial x_i}$

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_0}\left(\left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{X}} J + f\left.\frac{\partial J}{\partial t}\right|_{\boldsymbol{X}}\right) dX^d = \int_{\widehat{\Omega}_0}\left(\frac{Df}{Dt}J + fJ\frac{\partial u_i}{\partial x_i}\right) dX^d \tag{B.6}$$

We can now transform back the RHS integral to the current spatial domain

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_0}\left(\frac{Df}{Dt} + f\frac{\partial u_i}{\partial x_i}\right) J dX^d = \int_{\widehat{\Omega}_t}\left(\frac{Df}{Dt} + f\frac{\partial u_i}{\partial x_i}\right) dx^d \tag{B.7}$$

Expanding the material derivative

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_t}\left(\frac{\partial f}{\partial t} + u_i\frac{\partial f}{\partial x_i} + f\frac{\partial u_i}{\partial x_i}\right) dx^d = \int_{\widehat{\Omega}_t}\left(\frac{\partial f}{\partial t} + \frac{\partial (u_i f)}{\partial x_i}\right) dx^d \tag{B.8}$$

By using Gauss theorem

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_t}\frac{\partial f}{\partial t}dx^d + \int_{\partial\widehat{\Omega}_t} f u_i n_i dx^{d-1} \tag{B.9}$$

Reynolds Transport Theorem, which in the above has been given for a scalar, applies to a tensor of any order. Thus to apply it to a first order tensor (vector g_k) replace f by g_k

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} f dx^d = \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}}\int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_t}\left(\frac{\partial f}{\partial t} + \frac{\partial (u_i f)}{\partial x_i}\right) dx^d \tag{B.10}$$

$$\frac{D}{Dt}\int_{\widehat{\Omega}(t)} g_k dx^d = \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}}\int_{\widehat{\Omega}(t)} g_k dx^d = \int_{\widehat{\Omega}_t}\left(\frac{\partial g_k}{\partial t} + \frac{\partial (u_i g_k)}{\partial x_i}\right) dx^d \tag{B.11}$$

B.1.2. Generalized Reynolds Transport Theorem (GRTT)

The transport theorem can be extended to an arbitrary control volume that it is not necessarily a material volume. The movement of the control volume is given by a mapping between fictitious material particles with material coordinates $\boldsymbol{chi}$ and the spatial coordinates $\boldsymbol{x}$. We denote as $\Omega(t)$ the arbitrary control volume that in time t occupies the region in the spatial domain Ω_t and in time $t + \Delta t$ occupies $\Omega_{t+\Delta t}$. We assume that fictitious particles of the arbitrary volume control move with velocity $\boldsymbol{w}$ and we define a fictitious material derivative by $\frac{\partial f}{\partial t}\Big|_{\boldsymbol{\chi}} = \frac{\partial f}{\partial t} + \boldsymbol{w}_i \frac{\partial f}{\partial x_i}$.

Following a similar procedure, we will arrive to a generalized transport theorem given by

$$\frac{\partial}{\partial t}\Big|_{\boldsymbol{\chi}} \int_{\Omega(t)} f dx^d = \int_{\Omega_t} \left(\frac{\partial f}{\partial t} + \frac{\partial (\boldsymbol{w}_i f)}{\partial x_i} \right) dx^d \tag{B.12}$$

A generalized transport theorem for a volume control or a reference system can be obtained if we make the following replacements

- derivatives with respect to time following material particles $\frac{D}{Dt} = \frac{\partial}{\partial t}\big|_{\boldsymbol{X}}$ by derivatives with respect to time following the reference system $\frac{\partial}{\partial t}\big|_{\boldsymbol{\chi}}$
- the velocity vector for a material particle $\boldsymbol{v}$ by the velocity vector for a fictitious material particle of the reference system $\boldsymbol{w}$

B.1.3. Transport Theorem ALE

In previous sections we apply the transport theorem to a material system $\widehat{\Omega}(t)$ and to a virtual material system $\Omega(t)$ or control volume

$$\frac{D}{Dt} \int_{\widehat{\Omega}(t)} f dx^d = \frac{\partial}{\partial t}\Big|_{\boldsymbol{X}} \int_{\widehat{\Omega}(t)} f dx^d = \int_{\widehat{\Omega}_t} \frac{\partial f}{\partial t} dx^d + \int_{\partial \widehat{\Omega}_t} f u_i n_i dx^{d-1} \tag{B.13}$$

$$\frac{\partial}{\partial t}\Big|_{\boldsymbol{\chi}} \int_{\Omega(t)} f dx^d = \int_{\Omega_t} \frac{\partial f}{\partial t} dx^d + \int_{\partial \Omega_t} f w_i n_i dx^{d-1} \tag{B.14}$$

We can consider the particular situation when the material system $\widehat{\Omega}(t)$ and the referential system $\Omega(t)$ occupy in a time t the same region in the spatial domain. When this happens $\widehat{\Omega}_t$ and Ω_t are coincident. Subtraction of the two expressions leads to

$$\frac{D}{Dt} \int_{\widehat{\Omega}(t)} f dx^d = \frac{\partial}{\partial t}\Big|_{\boldsymbol{X}} \int_{\widehat{\Omega}(t)} f dx^d = \frac{\partial}{\partial t}\Big|_{\boldsymbol{\chi}} \int_{\Omega(t)} f d\Omega + \int_{\partial \Omega_t} f (u_i - w_i) n_i dx^{d-1} \tag{B.15}$$

Appendix C

Resumen extendido en español

C.1. Introducción

En la actualidad los avances en la rama de la Mecánica de Fluidos pueden producirse por una de las tres vías de conocimiento: experimental, teórico-analítico y computacional. Las interacciones entre las distintas vías de conocimiento han evolucionado a lo largo del tiempo. La rama computacional de la Dinámica de Fluidos (denominada CFD por sus siglas en inglés) se inició en la primera mitad del siglo XX. El crecimiento del CFD y su repercusión en el desarrollo virtual de proyectos se ha visto fuertemente impulsado por la evolución en la potencia de cálculo disponible y por los grandes avances en el desarrollo de los métodos numéricos que los respaldan. Esta tesis está encuadrada en la vía computacional de estudio de la Dinámica de Fluidos y dentro de ella en el desarrollo de un tipo de métodos conocidos como métodos sin malla.

Los métodos numéricos sin malla han captado la atención de investigadores e ingenieros para modelar flujos de fluidos que no son fácilmente modelables con los métodos tradicionales con malla. De manera general, los métodos sin malla son actualmente seleccionados para cubrir aplicaciones donde concurren interfases entre fluidos y dominios geométricos complejos. Siendo más específicos pueden citarse los siguientes campos de aplicación: flujos de superficie libre, interacción fluido-estructura, fenómenos de sloshing en tanques, inundaciones y tsunamis, flujos volcánicos de lava, procesos de moldeo por fundición y lubricación entre otros.

En el estado actual de desarrollo de los métodos numéricos sin malla, existe un cierto consenso en señalar los grandes desafíos a resolver para incrementar su fiabilidad. Incrementar la precisión y mejorar la convergencia constituyen uno de los grandes desafíos de este tipo de esquemas numéricos. El grupo GMNI ha introducido mejoras en métodos con malla utilizando técnicas que fueron concebidas para el uso de métodos sin malla. En este trabajo la transferencia entre los dos grandes grupos de métodos numéricos se realiza en la dirección opuesta. La idea de este trabajo se basa en el

desarrollo de un método sin malla al que le sean fácilmente implementadas técnicas que se han probado exitosas en métodos de volúmenes finitos de alto orden. De entre estas técnicas destacamos la incorporación de solvers de Riemann aproximados, la aproximación ALE, técnicas de estabilización *a posteriori* y los modelos de turbulencia implícitos.

C.1.1. Objetivos

El objetivo principal de esta tesis es el desarrollo de nuevas formulaciones sin malla de alta precisión que superen los inconvenientes de los métodos sin malla de alto orden y aumenten las prestaciones de las formulaciones sin malla existentes. El nuevo algoritmo ha de servir para el desarrollo de nuevas y más eficientes herramientas de simulación para cálculos CFD.

El objetivo principal se concreta en varios logros parciales que se detallan en la siguiente lista añadiendo algunos detalles sobre la estrategia a seguir.

- Mejora de la precisión de la aproximación kernel: Para superar esta carencia de la aproximación kernel se propone el desarrollo de una formulación basada en la técnica de Mínimos Cuadrados Móviles (MLS por sus siglas en inglés)
- Técnica de estabilización en presencia de discontinuidades: Como técnica de estabilización se implementa una técnica de estabilización *a posteriori* MOOD adaptada a formulaciones sin malla.
- Implementación de condiciones de contorno: La formulación de alta precisión propuesta admite la consideración de un término de contorno que facilita la implementación de las condiciones de contorno.
- Generalización y comparación con otras formulaciones: La formulación sin malla propuesta en esta tesis fue derivada partiendo de las ecuaciones de gobierno en forma ALE y explotando las propiedades de la técnica MLS para realizar aproximaciones y conducir la integración de la forma débil. Como resultado se obtiene una forma semi-discretizada muy amplia. Asumiendo configuraciones particulares en la formulación propuesta se llega a la forma semi-discretizada de otras formulaciones publicadas en la bibliografía.
- Tratamiento de la turbulencia: Como primeros pasos hacia la consecución de una formulación sin malla de alta precisión apta para resolver flujos turbulentos, se analiza la idoneidad de la formulación propuesta para simular turbulencia con la técnica implícita de grandes remolinos (iLES, por sus siglas en inglés).

C.2. Un método SPH-ALE débilmente compresible

La simulación numérica de flujos incompresibles con métodos sin malla resulta muy atractiva para multitud de aplicaciones en la ingeniería civil, mecánica y naval. La

imposición de la condición de incompresibilidad en un código de simulación numérica puede hacerse de forma estricta o de una forma aproximada. La elección de una u otra estrategia impone grandes diferencias en el algoritmo del código. La aproximación débilmente compresible hace uso de una ecuación de estado que acopla la presión y la densidad. La densidad no asume un valor constante exacto, pero en cambio el débil acoplamiento entre la ecuación de conservación de la masa y cantidad de movimiento permiten que pueda utilizarse los esquemas de resolución numérico de flujo compresible. La aproximación débilmente compresible presenta ventajas en cuanto a la facilidad de implementación cuando se dispone de un solver compresible y su mayor punto débil se encuentra en la restricción del paso de tiempo temporal de una formulación explícita. En esta tesis se ha optado por la aproximación débilmente compresible para abordar el estudio de los flujos de líquidos.

Se propone un método SPH-ALE de alta precisión para resolver flujos de líquidos mediante la aproximación débilmente compresible. Tomando como punto de partida el método SPH-ALE con reconstrucción constante se implementa la aproximación débilmente compresible y se incrementa la precisión mediante la introducción de la técnica MLS y la estabilización *a posteriori* MOOD. Se han considerado dos ecuaciones de estado: la ecuación de estado de Tait y la ecuación de Tammann. La primera de ellas es una ecuación barotrópica que liga la presión con la densidad mientras la segunda es una ecuación de estado que relaciona presión, densidad y energía interna. La ecuación de Tait es la ecuación de estado mayoritariamente usada en métodos SPH débilmente compresibles mientras que la ecuación de Tammann permite el acoplamiento adicional de la ecuación de la energía con la conservación de la cantidad de movimiento.

El método ha sido validado en una colección de problemas amplia. En primer lugar, se han considerado problemas de Riemann en dominios 1D y 2D. La ganancia de precisión obtenida con la incorporación de la reconstrucción de alto orden es notoria. Dichos casos con presencia de discontinuidades han servido para mostrar las mejoras que conlleva la estabilización a posteriori MOOD. La estabilización *a posteriori* se activa donde se requiere su actuación. La formulación propuesta ha sido probada con distintas posiciones iniciales de las partículas y tanto en la versión Lagrangiana como Euleriana. La técnica MOOD ha sido implementada por primera vez en un código Lagrangiano sin malla para aproximación débilmente compresible. Entre los resultados obtenidos destaca la capacidad de la formulación de respetar la simetría radial de problemas aun cuando la técnica de estabilización sólo emplea la información de vecindad entre partículas.

Además de problemas de Riemann la formulación ha sido puesta a prueba con problemas teóricos como el Taylor-Green bidimensional. Los resultados obtenidos han permitido constatar como la formulación débilmente compresible reproduce la solución teórica incompresible. Los múltiples ensayos del problema de Taylor-Green con varios números de Reynolds han permitido confirmar el buen desempeño de la implementación de los términos viscosos con gradientes de velocidad aproximados por MLS.

Los siguientes problemas test introducen la presencia de paredes. Se ha realizado la

simulación de problemas unidireccionales con solución analítica (Couette, Poiuseuille) y bidimensionales como la cavidad de pared móvil. Con una implementación de paredes basado en partículas fantasma se ha podido comprobar como la formulación propuesta rinde alta precisión.

C.3. El método MLSPH-ALE

La incorporación de la técnica MLS en la formulación SPH-ALE fue utilizada para incrementar la reconstrucción de variables requerida por un esquema Godunov de alta precisión y también para el cálculo de los flujos viscosos. Sin embargo, la formulación base para las ecuaciones de Euler (sin reconstrucción y sin flujos viscosos) no requiere de la técnica MLS y equivale a otras formulaciones sin malla empleadas en la bibliografía.

En la formulación MLSPH-ALE en cambio la utilización de la técnica MLS desbanca a la formulación kernel y asume un papel central tanto en la aproximación de las derivadas espaciales como en la integración de la forma débil. La función kernel se mantiene en la aproximación MLS tan sólo como función de peso. La aproximación MLS tiene garantizada la consistencia polinómica hasta un cierto orden que depende de la base de polinomios considerada.

La formulación MLSPH-ALE propuesta ha sido obtenida partiendo de la ecuación diferencial en forma ALE, utilizando propiedades de la aproximación MLS y finalmente introduciendo aproximaciones de cuadratura puntual en las partículas para integrar la forma débil.

Entre las ventajas obtenidas con la formulación MLSPH-ALE con respecto a las formulaciones SPH-ALE se destacan las siguientes. La aproximación MLS tiene consistencia asegurada y además no requiere un número tan elevado de vecinos como la formulación kernel para reducir el error de cuadratura. La demanda computational del cálculo MLS por partícula es del orden del cálculo de la técnica de renormalización utilizada en la aproximación kernel para recuperar la consistencia polinómica de primer orden. Otra ventaja importante de la formulación MLSPH-ALE es que la propia derivación de la formulación considera un término adicional que facilita la integración de las condiciones de contorno sin requerir de las técnicas de partículas fantasma.

Para la validación del modelo MLSPH-ALE se han considerado el problema de Taylor-Green dimensional y el problema de flujo alrededor de un cilindro. Los dos problemas han sido calculados en versión ALE y añadiendo una técnica de reorganización de partículas. Tanto para flujo viscoso como para flujo laminar los resultados están en concordancia próxima con los obtenidos con métodos con malla de alta precisión.

C.4. El método MLSPH-ALE y sus relaciones con otras formulaciones de partículas

El método MLSPH-ALE implementado en esta tesis es un desarrollo que está inspirado en ideas de otras formulaciones sin malla y hace uso de diferentes técnicas desarrolladas con diversos fines en la comunidad científica. En vez de remarcar las particularidades del método para diferenciarlo de formulaciones existentes se realiza un estudio para poner de manifiesto las relaciones con otros métodos numéricos que comparten el mismo objetivo de resolver numéricamente las ecuaciones de la Dinámica de Fluidos en la forma más eficiente posible.

Contrariamente a la separación compartimentada entre métodos basados en malla y métodos sin malla, se sigue una aproximación que considera tres grupos de métodos numéricos según el tipo de subdivisión realizado del dominio geométrico. Atendiendo a ese criterio se aprecian tres grupos: los métodos que realizan una descomposición precisa del dominio, los métodos que realizan una descomposición difusa y los métodos que ponderan el peso de una partícula en un punto sólo según su distancia sin importar su distribución. Según este criterio podemos tomar como métodos representativos de cada uno de estos grupos el método de los volúmenes finitos, los métodos sin malla basados en volúmenes finitos y los métodos SPH respectivamente. La creación de un grupo intermedio facilita la portabilidad de técnicas entre métodos que en una primera aproximación parece que no tienen nada en común. Esta distinción en tres grupos apareció por primera vez en la comunidad dedicada a la astrofísica en donde se tienen acérrimos defensores de los métodos de malla Cartesiana adaptativa por un lado y métodos Lagrangianos SPH por otro.

Con la irrupción de los métodos SPH-ALE se fue abriendo camino la idea de que hay un extenso campo de desarrollo a medio camino entre los métodos con malla y los métodos SPH Lagrangianos. En las dos últimas décadas han ido apareciendo multitud de métodos sin malla que pueden ser situados en esta categoría intermedia. La formulación MLSPH-ALE propuesta en esta tesis pertenece claramente a este grupo intermedio. Dada la generalidad asumida en la derivación de la formulación MLSPH-ALE se ha realizado un estudio buscando obtener las formas semi-discretizadas de modelos existentes partiendo desde configuraciones particulares de la formulación MLSPH-ALE. Se han analizado los vínculos existentes con las formulaciones Finite Volume Particle Method, el método SPH-ALE de Vila, la formulación sin malla de Ivanova, la formulación sin malla de Hopkins y el método FPM de Oñate.

C.5. El método MLSPH-ALE como candidato para LES implícito

Dada la abundancia de flujos turbulentos de interés en la ingeniería se han realizado pruebas test para comprobar la idoneidad del esquema MLSPH-ALE para hacer uso de modelos de turbulencia implícitos. Una de las fortalezas del esquema es la capacidad

de captar el rango subinercial de la cascada de energía de Kolmogorov con una discretización moderada. Además, la formulación propuesta no requiere de un mecanismo controlador del grado de disipación artificial ni tampoco de un modelo Smagorinsky para la subescala de las partículas. Se han considerado dos problemas test siendo el primero para flujo turbulento compresible y el segundo para flujo turbulento incompresible.

El esquema MLSPH-ALE fue capaz de capturar el espectro de energía en el problema de decaimiento turbulento para flujo compresible. A medida que la resolución espacial se incrementa el espectro de energía se aproxima más al espectro de referencia obtenido con DNS publicado en la bibliografía. Adicionalmente el modelo predice correctamente la evolución de las variables termodinámicas implicadas.

Para el análisis de decaimiento turbulento en flujo incompresible se recurre al problema de Taylor-Green tridimensional tanto en régimen no viscoso como para números de Reynolds moderados. Los resultados obtenidos prueban que el esquema MLSPH-ALE es capaz de capturar con una adecuada resolución la transición desde el régimen laminar inicial al desencadenamiento del flujo turbulento. La simulación con diferentes resoluciones permite observar como la evolución de la energía cinética y tasa de disipación convergen hacia los perfiles de referencia obtenidos con DNS.

Los resultados obtenidos para los casos turbulentos considerados son prometedores, ya que se comparan favorablemente a resultados obtenidos con esquemas de alto orden de diferencias finitas como el WENO5. Sin embargo, se requieren pasos adicionales para alcanzar un esquema sin malla de alta precisión que permite el cálculo de flujos turbulentos generales. Dado que el modelo sólo ha sido puesto a prueba con disposición Cartesiana fija de las partículas, el efecto de la disposición irregular o el movimiento de las partículas no se ha tenido en consideración. Otra importante extensión del tratamiento de la turbulencia debiera atender a la turbulencia en presencia de paredes.

C.6. Conclusiones

Durante la presente tesis se ha implementado y analizado una formulación sin malla que soluciona algunos de los inconvenientes de los métodos actuales de simulación CFD. La formulación propuesta está expresada en un marco de referencia ALE, se apoya en los solvers de Riemann para el cálculo de interacciones entre partículas y hace uso de la técnica de Mínimos Cuadrados Móviles (MLS). Además, durante la derivación de la formulación se mantiene un término de contorno que facilita la imposición de las condiciones de contorno en el modelo discreto. Además, los resultados obtenidos para casos de validación rinden grados de precisión comparables a métodos CFD de simulación con malla.

Las conclusiones generales extraídas de la realización de este trabajo son:

- Se ha implementado una formulación sin malla de alta precisión con aproximación débilmente compresible extendiendo la formulación SPH-ALE existente con la técnica de Mínimos Cuadrados Móviles (MLS) y la técnica de estabilización a

posteriori (MOOD). La técnica de estabilización propuesta mejora los resultados obtenidos con otras formulaciones sin malla que hacen uso de limitadores.

- La formulación MLSPH-ALE representa un paso adelante en la formulación de los métodos ALE sin malla existentes. El reemplazo de la aproximación MLS por la aproximación kernel garantiza la consistencia polinómica con orden determinado y proporciona ventajas adicionales para la integración de la forma débil y tratamiento de los términos viscosos.

- La formulación MLSPH-ALE es una formulación sin malla generalista que bajo configuraciones particulares contiene otras formulaciones publicadas en la bibliografía.

- La formulación MLSPH-ALE es idónea para acometer el cálculo de flujos turbulentos con modelación implícita de grandes remolinos. El uso de solvers de Riemann en la formulación MLSPH-ALE permite resolver problemas turbulentos con una estrategia ILES. Los resultados obtenidos en problemas de decaimiento para turbulencia compresible y turbulencia incompresible se comparan favorablemente con los resultados obtenidos con alguno de los modelos de diferencias finitas de alto orden publicados en la literatura.

C.6.1. Líneas de investigación futuras

Aunque el esquema sin malla propuesta aporta soluciones a algunos de los puntos débiles de las formulaciones actuales, hay todavía múltiples líneas de desarrollo de las formulaciones sin malla que non sido abordadas en esta tesis, como por ejemplo: modelos adicionales (multifase, turbulencia de pared, condiciones de contorno no-reflejantes, etc), eficiencia computacional, acometer la aplicación práctica a problemas particulares en la industria. Como líneas de extensión continuistas con el trabajo iniciado en esta tesis se destacan las siguientes:

- Implementación de una técnica de proyección para resolver flujos incompresibles: En el Capítulo 3 se acometió la resolución de flujos incompresibles con la aproximación débilmente compresible. Un esquema de proyección evita la restricción del paso de tiempo y también evita el precondicionamiento y algunas correcciones necesarias para flujos a muy bajos números de Mach. Como inconveniente, no pueden tratar discontinuidades y la implementación es más compleja.

- Implementación de una teselación local de Voronoi: El esquema MLSPH-ALE utiliza la misma molécula de vecinos para el cálculo de las funciones de forma MLS que para el sumatorio de flujos sobre los vecinos. Una teselación local para cada partícula podría ser de utilidad para el cálculo de flujos sobre un número reducido de vecinos y podría ser de ayuda para implementar una técnica de estabilización MOOD más próxima a la utilizada en métodos con malla.

- Tratamiento de choques en presencia de paredes: En el Capítulo 3 la técnica MOOD fue aplicada en la propagación de ondas de choque de pequeña amplitud en líquidos en dominios 1D y 2D sin presencia de paredes. En el Capítulo 4 la condición de contorno de pared fue impuesta sobre un cilindro. Sin embargo, no se ha considerado ningún problema donde la onda de choque interactúe con una pared en geometrías complejas.

- Acoplamiento de FVM-MLS y MLSPH-ALE: En la Tesis se ha resaltado la similitud del método sin malla MLSPH-ALE con el método de malla de volúmenes finitos FVM. Esta característica sugiere la idea de implementar una estrategia de acoplamiento entre ambas formulaciones. Los flujos de capa límite podrían ser resueltos muy convenientemente con esta estrategia de acoplamiento.

- Flujos de Superficie Libre (FSF) e Interacción Fluido-Estructura (FSI): Existen multitud de aplicaciones que demandan la extensión de métodos sin malla para abordar fenómenos con FSF y FSI. Por ejemplo, la generación de electricidad con Energía Eólica Marina es un sector altamente competitivo que demanda diseños eficientes capaces de hacer frente a los fenómenos FSI inducidos por violentos FSL. Inundaciones, tsunamis y volcanes son ejemplos de desastres naturales cuyos daños podrían reducirse si las zonas de riesgo disponen de modelos virtuales que permitan realizar actuaciones anticipándose a los acontecimientos.

Bibliography

Adami, S., Hu, X. Y., & Adams, N. A. (2012a). A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21), 7057–7075. ↑44

Adami, S., Hu, X. Y., & Adams, N. A. (2012b). Simulating three-dimensional turbulence with SPH. In *Proceedings of the Summer Program* (pp. 177–185). Center for Turbulence Research, CA. ↑132

Adami, S., Hu, X. Y., & Adams, N. A. (2013). A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241, 292–307. ↑96

Alonso-Asensio, I. (2019). Mesh-free hydrodynamics: theory and validation of the method. Master's thesis, Universidad de La Laguna. ↑109 , ↑120

Anderson Jr., J. D. (2010). Brief History of the Early Development of Theoretical and Experimental Fluid Dynamics. In *Encyclopedia of Aerospace Engineering*. John Wiley & Sons, Ltd. ↑1

Antoci, C., Gallati, M., & Sibilla, S. (2007). Numerical simulation of fluid–structure interaction by SPH. *Computers & Structures*, 85(11-14), 879–890. ↑143

Antuono, M., Sun, P. N., Marrone, S., & Colagrossi, A. (2021). The δ-ALE-SPH model: an arbitrary lagrangian-eulerian framework for the δ-SPH model with particle shifting technique. *Computers & Fluids*, 216, 104806. ↑97

Atluri, S. N. & Zhu, T. (1998). A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22(2), 117–127. ↑8

Avesani, D., Dumbser, M., & Bellin, A. (2014). A new class of Moving-Least-Squares WENO–SPH schemes. *Journal of Computational Physics*, 270, 278–299. ↑11 , ↑53 , ↑58 , ↑76 , ↑83

Avesani, D., Herrera, P., Chiogna, G., Bellin, A., & Dumbser, M. (2015). Smooth Particle Hydrodynamics with nonlinear Moving-Least-Squares WENO reconstruction to model anisotropic dispersion in porous media. *Advances in Water Resources*, 80, 43–59. ↑35

Bajko, J. (2013). Meshfree methods for computational aeroacoustics. Master's thesis, Brno University of technology. ↑83 , ↑126

Bajko, J., Čermák, L., & Jícha, M. (2014). High order finite point method for the solution to the sound propagation problems. *Computer Methods in Applied Mechanics and Engineering*, 280, 157–175. ↑126

Basic, J. (2021). *Development of numerical model for green water loading by coupling the mesh based flow models with meshless methods*. PhD thesis, University of Zagreb. ↑XI , ↑2

Bauer, A. & Springel, V. (2012). Subsonic turbulence in smoothed particle hydrodynamics and moving-mesh simulations. *Monthly Notices of the Royal Astronomical Society*, 423(3), 2558–2578. ↑132

Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2), 229–256. ↑8

Ben Moussa, B. & Vila, J. P. (2000). Convergence of SPH Method for Scalar Nonlinear Conservation Laws. *SIAM Journal on Numerical Analysis*, 37(3), 863–887. ↑10 , ↑35 , ↑43 , ↑47

Bilotta, G., Hérault, A., Cappello, A., Ganci, G., & Negro, C. D. (2015). GPUSPH: A Smoothed Particle Hydrodynamics model for the thermal and rheological evolution of lava flows. *Geological Society Special Publication*, 426(1), 387–408. ↑5

Blazek, J. (2015). *Computational Fluid Dynamics : Principles and Applications*. Elseiver/Butterworth-Heinemann, third edition. ↑2 , ↑97

Bonet, J. & Kulasegaram, S. (2000). Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47(6), 1189–1214. ↑8

Bonet, J. & Lok, T.-S. L. (1999). Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180(1-2), 97–115. ↑53

Brachet, M. E., Meiron, D. I., Orszag, S. A., Nickel, B. G., Morf, R. H., & Frisch, U. (1983). Small-scale structure of the Taylor–Green vortex. *Journal of Fluid Mechanics*, 130, 411–452. ↑132 , ↑137

Brebbia, C. A., Telles, J. C. F., & Wrobel, L. C. (1984). *Boundary Element Techniques*. Springer. ↑3

Buresti, G. (2012). *Elements of Fluid Dynamics*. Imperial College Press. ↑65 , ↑66

Cercos-Pita, J. L., Dalrymple, R. A., & Herault, A. (2016). Diffusive terms for the conservation of mass equation in SPH. *Applied Mathematical Modelling*, 40(19-20), 8722–8736. ↑97

Chandran, R. J. & Salih, A. (2019). A modified equation of state for water for a wide range of pressure and the concept of water shock tube. *Fluid Phase Equilibria*, 483, 182–188. ↑50

Chassaing, J.-C., Khelladi, S., & Nogueira, X. (2013). Accuracy assessment of a high-order moving least squares finite volume method for compressible flows. *Computers & Fluids*, 71, 41–53. ↑XI , ↑92

Chen, J.-S., Hillman, M., & Chi, S.-W. (2017). Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics*, 143(4), 04017001. ↑54 , ↑96

Chen, Z., Zong, Z., Li, H. T., & Li, J. (2013). An investigation into the pressure on solid walls in 2D sloshing using SPH method. *Ocean Engineering*, 59, 129–141. ↑53

Chiron, L. (2017). *Couplage et améliorations de la méthode SPH pour traiter des écoulements à multi-échelles temporelles et spatiales*. PhD thesis, École Centrale de Nantes. ↑52 , ↑66

Chiron, L., de Leffe, M., Oger, G., & Touzé, D. L. (2019). Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. *Computer Physics Communications*, 234, 93–111. ↑6

Clain, S., Diot, S., & Loubère, R. (2011). A high-order finite volume method for systems of conservation laws—Multi-dimensional Optimal Order Detection (MOOD). *Journal of Computational Physics*, 230(10), 4028–4050. ↑55 , ↑56

Cleary, P. W., Savage, G., Ha, J., & Prakash, M. (2014). Flow analysis and validation of numerical modelling for a thin walled high pressure die casting using SPH. *Computational Particle Mechanics*, 1(3), 229–243. ↑5

Colagrossi, A., Bouscasse, B., Antuono, M., & Marrone, S. (2012). Particle packing algorithm for SPH schemes. *Computer Physics Communications*, 183(8), 1641–1653. ↑49 , ↑76 , ↑89

Colagrossi, A., Colicchio, G., Lugni, C., & Brocchini, M. (2010). A study of violent sloshing wave impacts using an improved SPH method. *Journal of Hydraulic Research*, 48(sup1), 94–104. ↑5

Colagrossi, A. & Landrini, M. (2003). Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2), 448–475. ↑98 , ↑106

Collé, A., Limido, J., & Vila, J.-P. (2019). An accurate multi-regime SPH scheme for barotropic flows. *Journal of Computational Physics*, 388, 561–600. ↑10 , ↑35 , ↑49 , ↑57

Constant, E., Favier, J., Meldi, M., Meliga, P., & Serre, E. (2017). An immersed boundary method in OpenFOAM : Verification and validation. *Computers & Fluids*, 157, 55–72. ↑XI , ↑94

Crespo, A. J. C., Gómez-Gesteira, M., & Dalrymple, R. A. (2007). Boundary Conditions Generated by Dynamic Particles in SPH Methods. *Computers, Materials & Continua*, 5(3), 173–184. ↑44

Cueille, P. V. (2005). *Modélisation par Smoothed Particle Hydrodynamicdes phénomènes de diffusion présents dans un écoulement.* PhD thesis, Institut National des Sciences Appliquées de Toulouse. ↑35

Cueto-Felgueroso, L., Colominas, I., Mosqueira, G., Navarrina, F., & Casteleiro, M. (2004). On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 60(9), 1475–1512. ↑12

Cueto-Felgueroso, L., Colominas, I., Nogueira, X., Navarrina, F., & Casteleiro, M. (2007). Finite volume solvers and Moving Least-Squares approximations for the compressible Navier–Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 196(45-48), 4712–4736. ↑54 , ↑88

Cummins, S. J. & Rudman, M. (1999). An SPH Projection Method. *Journal of Computational Physics*, 152(2), 584–607. ↑10 , ↑48

Dalrymple, R. A. & Rogers, B. D. (2006). Numerical modeling of water waves with the SPH method. *Coastal Engineering*, 53(2-3), 141–147. ↑133

Dehnen, W. & Aly, H. (2012). Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2), 1068–1082. ↑28

Denner, F. (2021). The Gilmore-NASG model to predict single-bubble cavitation in compressible liquids. *Ultrasonics Sonochemistry*, 70, 105307. ↑50

Di Mascio, A., Antuono, M., Colagrossi, A., & Marrone, S. (2017). Smoothed particle hydrodynamics method from a large eddy simulation perspective. *Physics of Fluids*, 29, 035102. ↑11 , ↑76 , ↑133

Dilts, G. A. (1999). Moving-least-squares-particle hydrodynamics I. Consistency and stability. *International Journal for Numerical Methods in Engineering*, 44(8), 1115–1155. ↑8 , ↑76 , ↑98

Dilts, G. A. (2000). Moving least-squares particle hydrodynamics II: conservation and boundaries. *International Journal for Numerical Methods in Engineering*, 48(10), 1503–1524. ↑8 , ↑76 , ↑98

Douillet-Grellier, T. (2019). *Étude comparative des méthodes d'origine particulaire SPH et LBM pour la simulation d'écoulements polyphasiques intermittents dans des conduites*. PhD thesis, Université Paris Saclay. ↑7

Dubois, F. (2001). Partial Riemann problem, boundary conditions and gas dynamics. In *Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to Large Scale Computations* (pp. 16–77). Nova Science Publishers, Inc. ↑44

Dumbser, M., Zanotti, O., Loubère, R., & Diot, S. (2014). A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278, 47–75. ↑56

Eirís, A., Ramírez, L., Fernández-Fidalgo, J., Couceiro, I., & Nogueira, X. (2021). SPH-ALE Scheme for Weakly Compressible Viscous Flow with a Posteriori Stabilization. *Water*, 13(3), 245. ↑35 , ↑77 , ↑89

English, A., Domínguez, J. M., Vacondio, R., Crespo, A. J. C., Stansby, P. K., Lind, S. J., Chiapponi, L., & Gómez-Gesteira, M. (2021). Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): application to tank sloshing, dam break and fish pass problems. *Computational Particle Mechanics*. https://doi.org/10.1007/s40571-021-00403-3. ↑44

Euler, L. (1757). Principes généraux du mouvement des fluides. *Académie Royale des Sciences et des Belles-Lettres de Berlin*, Mémories 11. ↑1

Fehn, N., Kronbichler, M., Munch, P., & Wall, W. A. (2022). Numerical evidence of anomalous energy dissipation in incompressible Euler flows: towards grid-converged results for the inviscid Taylor–Green problem. *Journal of Fluid Mechanics*, 932, A40. ↑137

Fernández-Fidalgo, J., Clain, S., Ramírez, L., Colominas, I., & Nogueira, X. (2020). Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids. *Computer Methods in Applied Mechanics and Engineering*, 360, 112782. ↑12

Fernandez-Gutierrez, D., Souto-Iglesias, A., & Zohdi, T. I. (2017). A hybrid Lagrangian Voronoi–SPH scheme. *Computational Particle Mechanics*, 5(3), 345–354. ↑6

Ferrand, M., Joly, A., Kassiotis, C., Violeau, D., Leroy, A., Morel, F.-X., & Rogers, B. D. (2017). Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D. *Computer Physics Communications*, 210, 29–44. ↑44

Ferrand, M., Laurence, D. R., Rogers, B. D., Violeau, D., & Kassiotis, C. (2013). Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids*, 71(4), 446–472. ↑66

Few, C. G., Dobbs, C., Pettitt, A., & Konstandin, L. (2016). Testing hydrodynamics schemes in galaxy disc simulations. *Monthly Notices of the Royal Astronomical Society*, 460(4), 4382–4396. ↑99

Fougeron, G. (2018). *Contribution to the improvement of meshless methods applied to continuum mechanics*. PhD thesis, Université Paris Saclay. ↑43

Fourtakas, G., Dominguez, J. M., Vacondio, R., & Rogers, B. D. (2019). Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Computers & Fluids*, 190, 346–361. ↑66

Fourtakas, G., Stansby, P. K., Rogers, B. D., & Lind, S. J. (2018). An Eulerian–Lagrangian incompressible SPH formulation (ELI-SPH) connected with a sharp interface. *Computer Methods in Applied Mechanics and Engineering*, 329, 532–552. ↑65 , ↑143

Frontiere, N., Raskin, C. D., & Owen, J. M. (2017). CRKSPH – A Conservative Reproducing Kernel Smoothed Particle Hydrodynamics Scheme. *Journal of Computational Physics*, 332, 160–209. ↑8

Gaburov, E. & Nitadori, K. (2011). Astrophysical weighted particle magnetohydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 414(1), 129–154. ↑98 , ↑109 , ↑120 , ↑121 , ↑123

Gaburro, E., Boscheri, W., Chiocchetti, S., Klingenberg, C., Springel, V., & Dumbser, M. (2020). High order direct Arbitrary-Lagrangian-Eulerian schemes on moving Voronoi meshes with topology changes. *Journal of Computational Physics*, 407, 109167. ↑99

Gallouët, T., Hérard, J.-M., & Seguin, N. (2002). Some recent finite volume schemes to compute Euler equations using real gas EOS. *International Journal for Numerical Methods in Fluids*, 39(12), 1073–1138. ↑55

Garmann, D. J. & Visbal, M. R. (2015). AFRL Contributions to the Third International Workshop on High-Order CFD Methods. In *3rd International Workshop on High-Order CFD Methods* Miami, FL. ↑138

Garnier, E., Adams, N., & Sagaut, P. (2009). *Large Eddy Simulation for Compressible Flows*. Springer. ↑134

Garoosi, F. & Shakibaeinia, A. (2020). An improved high-order ISPH method for simulation of free-surface flows and convection heat transfer. *Powder Technology*, 376, 668–696. ↑48

Ghia, U., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3), 387–411. ↑70 , ↑71

Gingold, R. A. & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181, 375–389. ↑4 , ↑7

Gomez-Gesteira, M., Rogers, B. D., Dalrymple, R. A., & Crespo, A. J. C. (2010). State-of-the-art of classical SPH for free-surface flows. *Journal of Hydraulic Research*, 48(sup1), 6–27. ↑5

Gotoh, H. & Khayyer, A. (2018). On the state-of-the-art of particle methods for coastal and ocean engineering. *Coastal Engineering Journal*, 60(1), 79–103. ↑11

Gotoh, H., Shibahara, T., & Sakai, T. (2001). Sub-particle-scale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering. *Computational Fluid Dynamics Journal*, 9(4), 339–347. ↑133

Green, M. D., Vacondio, R., & Peiró, J. (2019). A smoothed particle hydrodynamics numerical scheme with a consistent diffusion term for the continuity equation. *Computers & Fluids*, 179, 632–644. ↑9

Gu, Y. T. (2005). Meshfree methods and their comparisons. *International Journal of Computational Methods*, 02(04), 477–515. ↑7

Guilmineau, E. & Queutey, P. (2002). A numerical simulation of vortex shedding from an oscillating circrular cylinder. *Journal of Fluids and Structures*, 16(6), 773–794. ↑XI , ↑94

He, X. & Doolen, G. (1997). Lattice Boltzmann Method on Curvilinear Coordinates System: Flow around a Circular Cylinder. *Journal of Computational Physics*, 134(2), 306–315. ↑XI , ↑92

Hickel, S., Egerer, C. P., & Larsson, J. (2014). Subgrid-scale modeling for implicit large eddy simulation of compressible flows and shock-turbulence interaction. *Physics of Fluids*, 26(10), 106101. ↑134

Hietel, D., Steiner, K., & Struckmeier, J. (2000). A finite-volume particle method for compressible flows. *Mathematical Models and Methods in Applied Sciences*, 10(09), 1363–1382. ↑10 , ↑76 , ↑98 , ↑110 , ↑112 , ↑142

Hinz, P. M. (2020). Implementing a meshless finite mass scheme in the cosmological N-body code Gadget. Master's thesis, Ludwig-Maximilians-Universität München. ↑109 , ↑120

Hirsch, C. (2007). *Numerical Computation of Internal and External Flows : The Fundamentals of Computational Fluid Dynamics*. Elsevier/Butterworth-Heinemann, second edition. ↑2 , ↑26 , ↑97

Holm, D. D. (1999). Fluctuation effects on 3D Lagrangian mean and Eulerian mean fluid motion. *Physica D: Nonlinear Phenomena*, 133(1-4), 215–269. ↑132

Honein, A. E. & Moin, P. (2004). Higher entropy conservation and numerical stability of compressible turbulence simulations. *Journal of Computational Physics*, 201(2), 531–545. ↑135

Hopkins, P. F. (2015). A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society*, 450(1), 53–110. ↑6 , ↑10 , ↑11 , ↑47 , ↑84 , ↑95 , ↑96 , ↑98 , ↑109 , ↑120 , ↑121 , ↑122 , ↑123 , ↑133

Hu, X. Y. & Adams, N. A. (2007). An incompressible multi-phase SPH method. *Journal of Computational Physics*, 227(1), 264–278. ↑89

Hu, X. Y. & Adams, N. A. (2015). A SPH Model for Incompressible Turbulence. *Procedia IUTAM*, 18, 66–75. ↑11 , ↑76

Huang, T.-H., Wei, H., Chen, J.-S., & Hillman, M. C. (2019). RKPM2D: an open-source implementation of nodally integrated reproducing kernel particle method for solving partial differential equations. *Computational Particle Mechanics*, 7(2), 393–433. ↑101

Huerta, A., Belytschko, T., Fernández-Méndez, S., Rabczuk, T., Zhuang, X., & Arroyo, M. (2017). Meshfree Methods. In *Encyclopedia of Computational Mechanics Second Edition*. John Wiley & Sons, Ltd. ↑96

Hussaini, M. (1998). On large-eddy simulation of compressible flows. In *29th AIAA Fluid Dynamics Conference* Albuquerque, NM. ↑135

Inutsuka, S. (2002). Reformulation of Smoothed Particle Hydrodynamics with Riemann Solver. *Journal of Computational Physics*, 179(1), 238–267. ↑9 , ↑98

Ivanova, N., Justham, S., Chen, X., Marco, O. D., Fryer, C. L., Gaburov, E., Ge, H., Glebbeek, E., Han, Z., Li, X.-D., Lu, G., Marsh, T., Podsiadlowski, P., Potter, A., Soker, N., Taam, R., Tauris, T. M., van den Heuvel, E. P. J., & Webbink, R. F. (2013). Common envelope evolution: where we stand and how we can move forward. *The Astronomy and Astrophysics Review*, 21(1). ↑35 , ↑47 , ↑76 , ↑98 , ↑100 , ↑101 , ↑105 , ↑107 , ↑119

Ivings, M. J., Causon, D. M., & Toro, E. F. (1998). On Riemann solvers for compressible liquids. *International Journal for Numerical Methods in Fluids*, 28(3), 395–418. ↑57 , ↑58

Jahanbakhsh, E., Maertens, A., Quinlan, N. J., Vessaz, C., & Avellan, F. (2017). Exact finite volume particle method with spherical-support kernels. *Computer Methods in Applied Mechanics and Engineering*, 317, 102–127. ↑117

Jahanbakhsh, E., Vessaz, C., Maertens, A., & Avellan, F. (2016). Development of a Finite Volume Particle Method for 3-D fluid flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 298, 80–107. ↑117

Ji, Z., Stanic, M., Hartono, E. A., & Chernoray, V. (2018). Numerical simulations of oil flow inside a gearbox by Smoothed Particle Hydrodynamics (SPH) method. *Tribology International*, 127, 47–58. ↑6

Junk, M. (2003). Do Finite Volume Methods Need a Mesh? In *Meshfree Methods for Partial Differential Equations* (pp. 223–238). Springer. ↑75

Junk, M. & Struckmeier, J. (2000). Consistency analysis of mesh-free methods for conservation laws. *Mitteilungen der Gesellschaft für Angewandte Mathematik und Mechanik*, 24(2), 99–126. ↑110

Kansa, E. (1990). Multiquadrics— A scattered data approximation scheme with applications to computational fluid-dynamics— I surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19(8-9), 127–145. ↑8

Katz, A. J. (2009). *Meshless methods for computational fluid dynamics*. PhD thesis, Stanford University. ↑126

Keck, R. & Hietel, D. (2005). A projection technique for incompressible flow in the meshless finite volume particle method. *Advances in Computational Mathematics*, 23(1-2), 143–169. ↑11 , ↑110

Koshizuka, S. & Oka, Y. (1996). Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid. *Nuclear Science and Engineering*, 123(3), 421–434. ↑10

Koshizuka, S., Shibata, K., Kondo, M., & Matsunaga, T. (2018). *Moving Particle Semi-implicit Method. A meshfree Particle Method for Fluid Dynamics*. Elsevier/Academic-Press. ↑10

Kotov, D. V., Yee, H. C., Wray, A. A., Sjögreen, B., & Kritsuk, A. G. (2016). Numerical dissipation control in high order shock-capturing schemes for LES of low speed flows. *Journal of Computational Physics*, 307, 189–202. ↑134

Koukouvinis, P., Kyriazis, N., & Gavaises, M. (2018). Smoothed particle hydrodynamics simulation of a laser pulse impact onto a liquid metal droplet. *PLoS ONE*, 13(9), e0204125. ↑141

Koukouvinis, P. K., Anagnostopoulos, J. S., & Papantonis, D. E. (2013). An improved MUSCL treatment for the SPH-ALE method: comparison with the standard SPH method for the jet impingement case. *International Journal for Numerical Methods in Fluids*, 71(9), 1152–1177. ↑57

Krimi, A., Ramírez, L., Khelladi, S., Navarrina, F., Deligant, M., & Nogueira, X. (2020). Improved δ-SPH Scheme with Automatic and Adaptive Numerical Dissipation. *Water*, 12(10), 2858. ↑VIII , ↑90 , ↑133

Kroll, N. (2006). ADIGMA - A European project on the development of adaptive higher-order variational methods for aerospace applications. In *European Conference on Computational Fluid Dynamics ECCOMAS CFD* TU Delft, The Netherlands: TU Delft. ↑3

Kuhnert, J. (2003). An Upwind Finite Pointset Method (FPM) for Compressible Euler and Navier-Stokes Equations. In *Lecture Notes in Computational Science and Engineering* (pp. 239–249). Springer. ↑99 , ↑126

Kulasegaram, S., Bonet, J., Lewis, R. W., & Profit, M. (2004). A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics*, 33(4), 316–325. ↑43

Kundu, P. K., Cohen, I., & Dowling, D. R. (2015). *Fluid Mechanics*. Elsevier/Academic-Press, sixth edition. ↑61 , ↑131

Lamichhane, B. P. (2001). The applications of Finite Volume Particle Method for Moving Boundary. Master's thesis, Universität Kaiserslautern. ↑110

Lancaster, P. & Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155), 141–158. ↑54 , ↑76

Lanson, N. & Vila, J.-P. (2008a). Renormalized Meshfree Schemes I: Consistency, Stability, and Hybrid Methods for Conservation Laws. *SIAM Journal on Numerical Analysis*, 46(4), 1912–1934. ↑35 , ↑98 , ↑121 , ↑125

Lanson, N. & Vila, J.-P. (2008b). Renormalized Meshfree Schemes II: Convergence for Scalar Conservation Laws. *SIAM Journal on Numerical Analysis*, 46(4), 1935–1964. ↑98

Lee, E.-S., Moulinec, C., Xu, R., Violeau, D., Laurence, D., & Stansby, P. (2008). Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics*, 227(18), 8417–8436. ↑71

Lee, S., Lele, S. K., & Moin, P. (1991). Eddy shocklets in decaying compressible turbulence. *Physics of Fluids A: Fluid Dynamics*, 3(4), 657–664. ↑134

LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press. ↑39

Li, C.-G. & Tsubokura, M. (2017). An implicit turbulence model for low-Mach Roe scheme using truncated Navier–Stokes equations. *Journal of Computational Physics*, 345, 462–474. ↑133

Li, Z. (2013). *Développement d'une méthode de simulation de couplage fluide-structure à l'aide de la méthode SPH*. PhD thesis, École Centrale de Lyon. ↑45 , ↑143

Li, Z., Leduc, J., Combescure, A., & Leboeuf, F. (2014). Coupling of SPH-ALE method and finite element method for transient fluid–structure interaction. *Computers & Fluids*, 103, 6–17. ↑45

Libersky, L. D., Petschek, A. G., Carney, T. C., Hipp, J. R., & Allahdadi, F. A. (1993). High Strain Lagrangian Hydrodynamics: A Three-Dimensional SPH Code for Dynamic Material Response. *Journal of Computational Physics*, 109(1), 67–75. ↑7

Lipnikov, K. & Shashkov, M. (2010). A framework for developing a mimetic tensor artificial viscosity for Lagrangian hydrocodes on arbitrary polygonal meshes. *Journal of Computational Physics*, 229(20), 7911–7941. ↑43

Liu, C. & Hu, C. (2014). An efficient immersed boundary treatment for complex moving object. *Journal of Computational Physics*, 274, 654–680. ↑XI , ↑94

Liu, C., Zheng, X., & Sung, C. H. (1998). Preconditioned Multigrid Methods for Unsteady Incompressible Flows. *Journal of Computational Physics*, 139(1), 35–57. ↑XI , ↑94

Liu, G. R. & Liu, M. B. (2003). *Smoothed Particle Hydrodynamics*. World Scientific. ↑7

Liu, M. & Zhang, Z. (2019). Smoothed particle hydrodynamics (SPH) for modeling fluid-structure interactions. *Science China Physics, Mechanics & Astronomy*, 62(8). ↑5

Liu, M. B. & Liu, G. R. (2006). Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, 56(1), 19–36. ↑34

Liu, M. B., Liu, G. R., & Lam, K. Y. (2003a). Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 155(2), 263–284. ↑28

Liu, M. B., Liu, G. R., Lam, K. Y., & Zong, Z. (2003b). Smoothed particle hydrodynamics for numerical simulation of underwater explosion. *Computational Mechanics*, 30(2), 106–118. ↑141

Liu, M. B., Xie, W. P., & Liu, G. R. (2005). Modeling incompressible flows using a finite particle method. *Applied Mathematical Modelling*, 29(12), 1252–1270. ↑126

Liu, W. K., Hao, W., Chen, Y., Jun, S., & Gosz, J. (1997). Multiresolution reproducing kernel particle methods. *Computational Mechanics*, 20(4), 295–309. ↑76

Liu, W. K., Jun, S., & Zhang, Y. F. (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9), 1081–1106. ↑8

Liu, W. T., Sun, P. N., Ming, F. R., & Zhang, A. M. (2018). Application of particle splitting method for both hydrostatic and hydrodynamic cases in SPH. *Acta Mechanica Sinica*, 34(4), 601–613. ↑XI , ↑94

Löhner, R. & Oñate, E. (1998). An advancing front point generation technique. *Communications in Numerical Methods in Engineering*, 14(12), 1097–1108. ↑4

Löhner, R., Sacco, C., Oñate, E., & Idelsohn, S. (2001). A finite point method for compressible flow. *International Journal for Numerical Methods in Engineering*, 53(8), 1765–1779. ↑126

Lu, X.-Y. & Dalton, C. (1996). Calculation of the timing of vortex formation from an oscillating cylinder. *Journal of Fluids and Structures*, 10(5), 527–541. ↑XI , ↑94

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82(12), 1013–1024. ↑4 , ↑7

Magoules, F. (2011). *Computational Fluid Dynamics*. Taylor and Francis/CRC Press. ↑34

Mancip, M. (2001). *Couplage de méthodes numériques pour les lois de conservation. Application au cas de l'injection.* PhD thesis, Institut National des Sciences Appliquées de Toulouse. ↑35

Marongiu, J.-C. (2007). *Méthode numérique lagrangienne pour la simulation d'écoulements à surface libre - Application aux turbines Pelton.* PhD thesis, École Centrale de Lyon. ↑6 , ↑10 , ↑35 , ↑44 , ↑57 , ↑109

Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Touzé, D. L., & Graziani, G. (2011). δ-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16), 1526–1542. ↑9 , ↑44 , ↑49 , ↑53 , ↑67

Marrone, S., Colagrossi, A., Antuono, M., Colicchio, G., & Graziani, G. (2013). An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers. *Journal of Computational Physics*, 245, 456–475. ↑XI , ↑92 , ↑94

Marrone, S., Mascio, A. D., & Touzé, D. L. (2016). Coupling of Smoothed Particle Hydrodynamics with Finite Volume method for free-surface flows. *Journal of Computational Physics*, 310, 161–180. ↑6

Mayrhofer, A., Laurence, D., Rogers, B. D., & Violeau, D. (2015). DNS and LES of 3-D wall-bounded turbulence using Smoothed Particle Hydrodynamics. *Computers & Fluids*, 115, 86–97. ↑11 , ↑76

Mayrhofer, A., Rogers, B. D., Violeau, D., & Ferrand, M. (2013). Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Computer Physics Communications*, 184(11), 2515–2527. ↑43 , ↑44

Michel-Dansac, V. (2016). *Development of high-order well-balanced schemes for geophysical flows*. PhD thesis, Université de Nantes. ↑39

Monaghan, J. J. (1988). An introduction to SPH. *Computer Physics Communications*, 48(1), 89–96. ↑98

Monaghan, J. J. (1989). On the problem of penetration in particle methods. *Journal of Computational Physics*, 82(1), 1–15. ↑9 , ↑96

Monaghan, J. J. (1992). Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1), 543–574. ↑9

Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2), 399–406. ↑5 , ↑9 , ↑44 , ↑48

Monaghan, J. J. (1997). SPH and Riemann Solvers. *Journal of Computational Physics*, 136(2), 298–307. ↑9

Monaghan, J. J. (2002). SPH compressible turbulence. *Monthly Notices of the Royal Astronomical Society*, 335(3), 843–852. ↑11 , ↑76 , ↑132

Monaghan, J. J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8), 1703–1759. ↑9 , ↑28 , ↑98 , ↑105

Monaghan, J. J. (2011). A turbulence model for Smoothed Particle Hydrodynamics. *European Journal of Mechanics - B/Fluids*, 30(4), 360–370. ↑11 , ↑76

Monaghan, J. J. (2012). Smoothed Particle Hydrodynamics and Its Diverse Applications. *Annual Review of Fluid Mechanics*, 44(1), 323–346. ↑132

Monaghan, J. J. & Gingold, R. A. (1983). Shock simulation by the particle method SPH. *Journal of Computational Physics*, 52(2), 374–389. ↑9

Monaghan, J. J. & Kajtar, J. B. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10), 1811–1820. ↑44

Monaghan, J. J. & Lattanzio, J. C. (1985). A refined particle method for astrophysical problems. *Astonomics and Astrophysics*, 149(1), 135–143. ↑29

Morris, J. P. (1996). A study of the stability properties of smooth particle hydrodynamics. *Publications of the Astronomical Society of Australia*, 13(1), 97–102. ↑52 , ↑77

Morris, J. P., Fox, P. J., & Zhu, Y. (1997). Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1), 214–226. ↑9 , ↑44 , ↑65 , ↑66

Muñoz, D. J., Springel, V., Marcus, R., Vogelsberger, M., & Hernquist, L. (2012). Multidimensional, compressible viscous flow on a moving Voronoi mesh. *Monthly Notices of the Royal Astronomical Society*, 428(1), 254–279. ↑97

Navier, C.-L. M. H. (1823). Mémoire sur les lois du mouvement des fluides. *Mémoires de L'academie Royales des sciences de L'institut de France*, VI, 389–440. ↑1

Nayroles, B., Touzot, G., & Villon, P. (1992). Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5), 307–318. ↑8

Nestor, R. M., Basa, M., Lastiwka, M., & Quinlan, N. J. (2009). Extension of the finite volume particle method to viscous flow. *Journal of Computational Physics*, 228(5), 1733–1749. ↑11 , ↑76 , ↑79 , ↑89 , ↑110

Neuhauser, M. (2014). *Development of a coupled SPH-ALE Finite Volume method for the simulation of transient flows in hydraulic machines*. PhD thesis, École Centrale de Lyon. ↑45

Ng, Y. T., Min, C., & Gibou, F. (2009). An efficient fluid–solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228(23), 8807–8829. ↑XI , ↑94

Niu, X. D., Chew, Y. T., & Shu, C. (2003). Simulation of flows around an impulsively started circular cylinder by Taylor series expansion- and least squares-based lattice Boltzmann method. *Journal of Computational Physics*, 188(1), 176–193. ↑XI , ↑92

Nogueira, X., Cueto-Felgueroso, L., Colominas, I., Navarrina, F., & Casteleiro, M. (2010). A new shock-capturing technique based on Moving Least Squares for higher-order numerical schemes on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40), 2544–2558. ↑12

Nogueira, X., Ramírez, L., Clain, S., Loubère, R., Cueto-Felgueroso, L., & Colominas, I. (2016a). High-accurate SPH method with Multidimensional Optimal Order Detection limiting. *Computer Methods in Applied Mechanics and Engineering*, 310, 134–155. ↑11 , ↑35 , ↑53 , ↑55 , ↑58 , ↑76

Nogueira, X., Ramírez, L., Khelladi, S., Chassaing, J.-C., & Colominas, I. (2016b). A high-order density-based finite volume method for the computation of all-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 298, 229–251. ↑48

Oger, G. (2006). *Aspects théoriques de la méthode SPH et applications à l'hydrodynamique à surface libre*. PhD thesis, École Centrale de Nantes. ↑52

Oger, G., Marrone, S., Touzé, D. L., & de Leffe, M. (2016). SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms. *Journal of Computational Physics*, 313, 76–98. ↑10 , ↑35 , ↑47 , ↑97

Oger, G., Vergnaud, A., Bouscasse, B., Ohana, J., Zarim, M. A., Leffe, M. D., Bannier, A., Chiron, L., Jus, Y., Garnier, M., Halbout, S., & Touzé, D. L. (2020). Simulations of helicopter ditching using smoothed particle hydrodynamics. *Journal of Hydrodynamics*, 32(4), 653–663. ↑6

Oñate, E. (1998). Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems. *Computer Methods in Applied Mechanics and Engineering*, 151(1-2), 233–265. ↑76 , ↑125

Oñate, E., Idelsohn, S., Zienkiewicz, O., Taylor, R., & Sacco, C. (1996a). A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4), 315–346. ↑8 , ↑125

Oñate, E., Idelsohn, S., Zienkiewicz, O. C., & Taylor, R. L. (1996b). A finite point method in computational mechanics. applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 39(22), 3839–3866. ↑76 , ↑99 , ↑125

Ortega, E. (2014). *Development and applications of the Finite Point Method to compressible aerodynamics problems*. PhD thesis, Universitat Politècnica de Catalunya. ↑126

Ortega, E., Oñate, E., & Idelsohn, S. (2009). A finite point method for adaptive three-dimensional compressible flow calculations. *International Journal for Numerical Methods in Fluids*, 60(9), 937–971. ↑76 , ↑126

Paillère, H., Corre, C., & Cascales, J. R. G. (2003). On the extension of the AUSM scheme to compressible two-fluid models. *Computers & Fluids*, 32(6), 891–916. ↑51

Parshikov, A. N. & Medin, S. A. (2002). Smoothed Particle Hydrodynamics Using Interparticle Contact Algorithms. *Journal of Computational Physics*, 180(1), 358–382. ↑9 , ↑98

Pineda, S., Marongiu, J.-C., Aubert, S., & Lance, M. (2019). Simulation of a gas bubble compression in water near a wall using the SPH-ALE method. *Computers & Fluids*, 179, 459–475. ↑58 , ↑141

Poinsot, T. J. & Lele, S. K. (1992). Boundary Conditions for Direct Simulations of Compressible Viscous Flows. *Journal of Computational Physics*, 101(1), 104–129. ↑45

Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press. ↑VIII , ↑131 , ↑132

Price, D. J. (2004). *Magnetic fields in Astrophysics*. PhD thesis, University of Cambridge. ↑28

Price, D. J. (2011). Resolving high Reynolds numbers in smoothed particle hydrodynamics simulations of subsonic turbulence. *Monthly Notices of the Royal Astronomical Society: Letters*, 420(1), L33–L37. ↑132

Price, D. J. (2012). Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3), 759–794. ↑105

Quinlan, N. & Lobovský, L. (2018). The Finite Volume Particle Method. In *Numerical Methods and Advanced Simulation in Biomechanics and Biological Processes* (pp. 341–354). Elsevier. ↑117

Quinlan, N. J. (2018). Extensions of the meshless finite volume particle method (FVPM) for static and dynamic free-surface flows. *Computers & Fluids*, 177, 33–45. ↑11 , ↑110

Quinlan, N. J., Lobovský, L., & Nestor, R. M. (2014). Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area. *Computer Physics Communications*, 185(6), 1554–1563. ↑115 , ↑117

Rajani, B. N., Kandasamy, A., & Majumdar, S. (2009). Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3), 1228–1247. ↑XI , ↑94

Ramírez, L., Nogueira, X., Khelladi, S., Chassaing, J.-C., & Colominas, I. (2014). A new higher-order finite volume method based on Moving Least Squares for the resolution of the incompressible Navier–Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 278, 883–901. ↑12 , ↑48 , ↑88 , ↑97

Ramírez, L., Nogueira, X., Khelladi, S., Krimi, A., & Colominas, I. (2018). A very accurate Arbitrary Lagrangian–Eulerian meshless method for Computational Aeroacoustics. *Computer Methods in Applied Mechanics and Engineering*, 342, 116–141. ↑76 , ↑77 , ↑87 , ↑98

Ramírez, L., Nogueira, X., Ouro, P., Navarrina, F., Khelladi, S., & Colominas, I. (2017). A Higher-Order Chimera Method for Finite Volume Schemes. *Archives of Computational Methods in Engineering*, 25(3), 691–706. ↑VIII , ↑XI , ↑12 , ↑86 , ↑91 , ↑92

Randles, P. W. & Libersky, L. D. (1996). Smoothed Particle Hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4), 375–408. ↑7 , ↑8

Renaut, G.-A. (2015). *Schémas d'ordre élevé pour la méthode SPH-ALE appliquée à des simulations sur machines hydrauliques*. PhD thesis, École Centrale de Lyon. ↑35

Rizzetta, D., Visbal, M., & Blaisdell, G. (1999). Application of a high-order compact difference scheme to large-eddy and direct numerical simulation. In *30th AIAA Fluid Dynamics Conference* Norfolk, VA: AIAA American Institute of Aeronautics and Astronautics. ↑134

Robinson, M. & Monaghan, J. J. (2011). Direct numerical simulation of decaying two-dimensional turbulence in a no-slip square box using smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 70(1), 37–55. ↑132

Rosswog, S. (2009). Astrophysical smooth particle hydrodynamics. *New Astronomy Reviews*, 53(4-6), 78–104. ↑105

Rusanov, V. V. (1962). The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1(2), 304–320. ↑87

Salsa, S. (2016). *Partial Differential Equations in Action. From Modelling to Theory.* Springer, third edition. ↑39

Sarkar, S., Erlebacher, G., Hussaini, M. Y., & Kreiss, H. O. (1991). The analysis and modelling of dilatational terms in compressible turbulence. *Journal of Fluid Mechanics*, 227, 473–493. ↑134

Saurel, R., Cocchi, J. P., & Butler, P. B. (1999). Numerical Study of Cavitation in the Wake of a Hypervelocity Underwater Projectile. *Journal of Propulsion and Power*, 15(4), 513–522. ↑51

Schaller, M., Bower, R. G., & Theuns, T. (2013). On the use of particle based methods for cosmological hydrodynamical simulations. In *8th International SPHERIC workshop* Trondheim, Norway. ↑76 , ↑99

Schiesser, W. E. (1991). *The Numerical Method of Lines. Integration of Partial Differential Equations.* Academic Press. ↑26

Schiesser, W. E. & Griffiths, G. W. (2009). *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab.* Cambridge University Press. ↑26

Schnabel, D., Özkaya, E., Biermann, D., & Eberhard, P. (2018). Modeling the motion of the cooling lubricant in drilling processes using the finite volume and the smoothed particle hydrodynamics methods. *Computer Methods in Applied Mechanics and Engineering*, 329, 369–395. ↑5

Schwartzkopff, T., Dumbser, M., & Munz, C.-D. (2004). Fast high order ADER schemes for linear hyperbolic equations. *Journal of Computational Physics*, 197(2), 532–539. ↑3

Schwartzkopff, T., Munz, C. D., & Toro, E. F. (2002). ADER: A High-Order Approach for Linear Hyperbolic Systems in 2D. *Journal of Scientific Computing*, 17(1/4), 231–240. ↑3

Scovazzi, G. & Hughes, T. J. R. (2007). *Lecture Notes on Continuum Mechanics on Arbitrary Moving Domains*. Technical Report Report 2007-6312P, Sandia National Laboratories. ↑16

Shao, S. (2006). Simulation of breaking wave by SPH method coupled with k-ϵ model. *Journal of Hydraulic Research*, 44(3), 338–349. ↑133

Shao, S. & Gotoh, H. (2005). Turbulence particle models for tracking free surfaces. *Journal of Hydraulic Research*, 43(3), 276–289. ↑133

Sherwin, S. J. & Peiró, J. (2001). Mesh generation in curvilinear domains using high-order elements. *International Journal for Numerical Methods in Engineering*, 53(1), 207–223. ↑3 , ↑4

Shu, C.-W., Don, W.-S., Gottlieb, D., Schilling, O., & Jameson, L. (2005). Numerical Convergence Study of Nearly Incompressible, Inviscid Taylor–Green Vortex Flow. *Journal of Scientific Computing*, 24(1), 1–27. ↑137 , ↑138

Sjah, J. (2013). *Couplage SPH -DEM pour l'étude de l'érosion dans les ouvrages hydrauliques*. PhD thesis, École Centrale de Lyon. ↑52

Springel, V. (2010). E pur si muove:galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 401(2), 791–851. ↑97 , ↑105

Spyropoulos, E. T. & Blaisdell, G. A. (1996). Evaluation of the dynamic model for simulations of compressible decaying isotropic turbulence. *AIAA Journal*, 34(5), 990–998. ↑134

Stokes, G. G. (1845). On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solids. *Cambridge Philosophical Transactions*, VII(I), 287–319. ↑1

Suchde, P. (2018). *Conservation and accuracy in meshfree generalized finite difference methods*. PhD thesis, Universität Kaiserslautern. ↑126

Suchde, P., Kuhnert, J., Schröder, S., & Klar, A. (2017). A flux conserving meshfree method for conservation laws. *International Journal for Numerical Methods in Engineering*, 112(3), 238–256. ↑99

Sun, P. N., Colagrossi, A., Marrone, S., Antuono, M., & Zhang, A.-M. (2019). A consistent approach to particle shifting in the δ-Plus-SPH model. *Computer Methods in Applied Mechanics and Engineering*, 348, 912–934. ↑VIII , ↑64 , ↑89 , ↑90 , ↑94

Sun, P. N., Colagrossi, A., Marrone, S., & Zhang, A. M. (2017). The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme. *Computer Methods in Applied Mechanics and Engineering*, 315, 25–49. ↑64

Sun, X., Zhang, J.-Z., & Ren, X.-L. (2012). Characteristic-Based Split (CBS) Finite Element Method for Incompressible Viscous Flow with Moving Boundaries. *Engineering Applications of Computational Fluid Mechanics*, 6(3), 461–474. ↑XI , ↑94

Szewc, K. (2013). *Development of Smoothed Particle Hydrodynamics approach for modelling of multiphase flows with interfaces.* PhD thesis, Université de Lorraine. ↑44

Szewc, K., Pozorski, J., & Minier, J.-P. (2012). Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 92(4), 343–369. ↑71

Taylor, G. I. & Green, A. E. (1937). Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 158(895), 499–521. ↑61 , ↑89

Teleaga, D. (2005). *A Finite-Volume Particle Method for Conservation Laws.* PhD thesis, Universität Hamburg. ↑110

Teleaga, D. & Struckmeier, J. (2008). A finite-volume particle method for conservation laws on moving domains. *International Journal for Numerical Methods in Fluids*, 58(9), 945–967. ↑110

Tennekes, H. (1972). *A first course in turbulence.* The MIT Press. ↑131

Toro, E. F. (2009). *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, third edition. ↑57 , ↑59

Trask, N., Perego, M., & Bochev, P. (2017). A High-Order Staggered Meshless Method for Elliptic Problems. *SIAM Journal on Scientific Computing*, 39(2), A479–A502. ↑43

Vacondio, R., Altomare, C., Leffe, M. D., Hu, X., Touzé, D. L., Lind, S., Marongiu, J.-C., Marrone, S., Rogers, B. D., & Souto-Iglesias, A. (2020). Grand challenges for smoothed particle hydrodynamics numerical schemes. *Computational Particle Mechanics*, 8, 575–588. ↑11 , ↑43

Vacondio, R., Rogers, B. D., Stansby, P. K., & Mignosa, P. (2013). Shallow water SPH for flooding with dynamic particle coalescing and splitting. *Advances in Water Resources*, 58, 10–23. ↑5

van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1), 101–136. ↑10 , ↑75

Vanella, M. & Balaras, E. (2009). A moving-least-squares reconstruction for embedded-boundary formulations. *Journal of Computational Physics*, 228(18), 6617–6628. ↑XI , ↑94

Vessaz, C. (2015). *Finite Particle Flow Simulation of Free Jet Deviation by Rotating Pelton Buckets*. PhD thesis, École Polytechnique Fédérale de Lausanne. ↑6 , ↑109

Vila, J. P. (1999). On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*, 09(02), 161–209. ↑XI , ↑10 , ↑15 , ↑35 , ↑36 , ↑39 , ↑43 , ↑47 , ↑57 , ↑75 , ↑76 , ↑83 , ↑98 , ↑142

Vila, J. P. (2005). SPH Renormalized Hybrid Methods for Conservation Laws: Applications to Free Surface Flows. In *Meshfree Methods for Partial Differential Equations II* (pp. 207–229). Springer. ↑35

Violeau, D. (2012). *Fluid Mechanics and the SPH Method*. Oxford University Press. ↑105

Violeau, D., Piccon, S., & Chabard, J. P. (2002). Two attempts of turbulence modelling in smoothed particle hydrodynamics. In *Advances in Fluid Modeling and Turbulence Measurements. Proceedings of the 8th International Symposium on Flow Modeling and Turbulence Measurements* (pp. 339–346). Tokyo, Japan: World Scientific. ↑133

Violeau, D. & Rogers, B. D. (2016). Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. *Journal of Hydraulic Research*, 54(1), 1–26. ↑11

Visbal, M. R. & Rizzetta, D. P. (2002). Large-Eddy Simulation on Curvilinear Grids Using Compact Differencing and Filtering Schemes. *Journal of Fluids Engineering*, 124(4), 836–847. ↑VIII , ↑135 , ↑136

Vittoz, L., Oger, G., de Leffe, M., & Touzé, D. L. (2019). Comparisons of weakly-compressible and truly incompressible approaches for viscous flow into a high-order Cartesian-grid finite volume framework. *Journal of Computational Physics: X*, 1, 100015. ↑61

Walhorn, E., Kölke, A., Hübner, B., & Dinkler, D. (2005). Fluid–structure coupling within a monolithic model involving free surface flows. *Computers & Structures*, 83(25-26), 2100–2111. ↑143

Wang, Z.-B., Chen, R., Wang, H., Liao, Q., Zhu, X., & Li, S.-Z. (2016). An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling*, 40(23), 9625 – 9655. ↑5

Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., & Visbal, M. (2013). High-order CFD methods: current status and

perspective. *International Journal for Numerical Methods in Fluids*, 72(8), 811–845. ↑3 , ↑4

Werdelmann, B., Koch, R., Krebs, W., & Bauer, H.-J. (2021). An approach for permeable boundary conditions in SPH. *Journal of Computational Physics*, 444, 110562. ↑45

Zienkiewicz, O. C., Taylor, R. L., & Nithiarasu, P. (2014). *The Finite Element Method for Fluid Dynamics*. Elsevier/Butterworth-Heinemann, seventh edition. ↑2 , ↑97