

# A single layer ghost particle boundary method for Smoothed Particle Hydrodynamics

Eine einschichtige Geisterpartikelgrenzenmethode  
für die Hydrodynamik geglätteter Partikel

Scientific work for obtaining the academic degree

Master of Science (M.Sc.) in Computational Science and Engineering

at the TUM School of Computation, Information and Technology of the Technical University of Munich

**Supervisor** PD. Dr.-Ing. habil. Stefan Adami  
Munich Institute of Integrated Materials, Energy and Process Engineering

**Submitted by** José María Cela París

**Submitted on** December 15, 2024 in Garching



## Abstract

This work introduces a single-layer ghost particle boundary method for Smoothed Particle Hydrodynamics (SPH), addressing the challenges of enforcing boundary conditions at static and moving solid interfaces. The proposed method combines principles from dummy particles, and fixed ghost particles to model fluid-solid interactions effectively. The method is formulated in terms of marker particles, placed along solid boundaries. The marker particles generate virtual particles based on the curvature and unit normal vector of the solid boundary surface. The virtual particles are used to compute the fluid-solid interaction forces and enforce the no-slip boundary condition. This formulation allows the method to accurately represent the geometry of solid boundaries, independently of resolution. The method incorporates advanced techniques for handling curvature, sharp corners, and thin walls, offering significant improvements in computational accuracy and versatility when discretizing solid boundaries. The proposed method naturally works in two and three dimensions and is implemented in a general-purpose SPH framework. Validation is performed against analytical solutions and diverse flow scenarios, demonstrating the robustness and versatility of the method in modeling fluid dynamics problems with intricate solid boundaries.

## Zusammenfassung

Diese Arbeit stellt eine Einzelschicht-Ghost-Particle-Grenzmethode für Smoothed Particle Hydrodynamics (SPH) vor, die sich den Herausforderungen bei der Durchsetzung von Randbedingungen an statischen und beweglichen Festkörpergrenzen widmet. Die vorgeschlagene Methode kombiniert Prinzipien von Dummy-Partikeln und festen Ghost-Partikeln, um Fluid-Festkörper-Interaktionen effektiv zu modellieren. Die Methode wird anhand von Marker-Partikeln formuliert, die entlang der Festkörpergrenzen platziert werden. Die Marker-Partikel erzeugen virtuelle Partikel basierend auf der Krümmung und dem Einheitsnormalenvektor der Festkörpergrenzfläche. Diese virtuellen Partikel werden verwendet, um die Fluid-Festkörper-Interaktionskräfte zu berechnen und die No-Slip-Randbedingung durchzusetzen. Diese Formulierung ermöglicht es der Methode, die Geometrie von Festkörpergrenzen unabhängig von der Auflösung präzise darzustellen. Die Methode integriert fortschrittliche Techniken zur Handhabung von Krümmungen, scharfen Ecken und dünnen Wänden und bietet bedeutende Verbesserungen in der rechnerischen Genauigkeit und Vielseitigkeit bei der Diskretisierung von Festkörpergrenzen. Die vorgeschlagene Methode funktioniert von Natur aus in zwei und drei Dimensionen und ist in ein allgemeines SPH-Framework implementiert. Die Validierung erfolgt durch analytische Lösungen und verschiedene Strömungsszenarien, wobei die Robustheit und Vielseitigkeit der Methode bei der Modellierung von Flüssigkeitsdynamikproblemen mit komplexen Festkörpergrenzen demonstriert wird.



# Contents

<b>List of Figures</b>	vii
<b>List of Tables</b>	ix
<b>Acronyms</b>	xi
<b>1 Introduction</b>	1
1.1 Introduction to Smoothed Particle Hydrodynamics . . . . .	2
<b>2 Method</b>	5
2.1 Governing Equations . . . . .	5
2.2 Numerical Method . . . . .	5
2.2.1 Density . . . . .	6
2.2.2 Momentum Equation . . . . .	7
2.3 Time Stepping . . . . .	7
<b>3 Solid Wall Modelling</b>	9
3.1 Discretization of the Solid Interface . . . . .	9
3.2 Volume of Virtual Particles . . . . .	10
3.3 Velocity and Pressure Mirroring . . . . .	12
3.4 Force and Density Computation . . . . .	13
3.5 Computing the Normal Vectors and Curvature . . . . .	14
3.6 Handling Sharp Corners . . . . .	15
3.7 Interaction Limiters for Submerged Thin Walls . . . . .	18
<b>4 Comparison With Analytical Solutions</b>	21
4.1 Poiseuille and Couette Flow . . . . .	21
4.2 Taylor-Couette Flow . . . . .	21
4.3 Convergence . . . . .	22
<b>5 Results</b>	25
5.1 Periodic Array of Cylinders . . . . .	25
5.2 Periodic Lattice of Cylinders . . . . .	27
5.3 Lid-Driven Cavity . . . . .	29
5.4 Flow Around a Moving Square Inside a Rectangular Box . . . . .	31
5.5 Flow around an elliptical and triangular cylinder . . . . .	34
5.6 Flow Over a Sphere . . . . .	38
5.7 Dam Break with Adjacent Hydrostatic Tank . . . . .	40
5.8 Flow Driven by Rotating Polar Roses . . . . .	42
<b>6 Conclusions</b>	45
<b>A Code</b>	47
<b>Bibliography</b>	49



# List of Figures

1.1	Quintic spline kernel, and kernel gradient. . . . .	3
3.1	Schematic representation of the virtual particles (grey) generated by a marker particle (black) interacting with a fluid particle (white). Relevant distances for the interaction are shown. . . . .	9
3.2	Different wall geometries discretized with marker particles (black). The virtual particles (grey) are generated in the opposite direction of the normal vector. Note that the curvature of the wall affects the distribution and volume of the virtual particles, as seen in the spacing and size differences between flat, circular, and sinusoidal geometries. . . . .	10
3.3	Volume of the virtual particles in a convex and concave scenario. Marker particles with their normal vectors represented in black. Length element $\Delta s$ highlighted in red. Circles dividing the virtual particles in layers, with radius $R_0$ dashed lines. It can be seen that for convex curvature the volume of the first layer of particles is larger than the second and third. The opposite is true for concave curvature. . . . .	11
3.4	Examples of numerically computed normal vectors, and curvature for smooth geometries. The normal vectors are represented in black, and the curvature by the color scale. . . . .	15
3.5	Volume of the three virtual particles, depending on the curvature/radius of curvature. Following Eq.3.24, setting $\Delta s = \Delta x$ . . . . .	16
3.6	Volume of the virtual particles at a sharp triangular corner, for different values of $\kappa_{\max}$ . Marker particles are represented in black, first second and third virtual particles are represented in red, green and blue respectively. . . . .	17
3.7	Examples of numerically computed normal vectors, and curvature for squares with concave and convex corners and an equilateral triangle. The squares have side $L = 2$ . The normal vectors are represented in black, and the curvature by the color scale. The particle spacing is equal to the marker spacing, $\Delta s = \Delta x = \frac{1}{15}$ and the maximum curvature is $\kappa_{\max} = 10$ . . . . .	17
3.8	Diagram illustrating the concept of the thin wall interaction limiter. Fluid particles (blue) close to a marker particles (black) are assigned the normal vector of the closest marker particle. A fluid particle (circled in red) interacts only with particles with aligned normal vectors (marked by squares). . . . .	18
3.9	Diagram illustrating the need to allow interaction of fluid particles with opposed normals in some problematic cases. If the interaction is limited only to particles with aligned normals, the fluid particles in the upper half of the duct would not interact with the ones at the lower half. . . . .	18
4.1	Poiseuille and Couette flow comparison between analytical solution and SPH solution. Time evolution from rest to steady state at $t = 0.2, 1, 2, 10$ . . . . .	22
4.2	Taylor-Couette . . . . .	23
4.3	$L_1$ error norm for analytical flows in the near wall region. As a function of the smoothing length $h$ . Lines with first and second order convergence rates are shown for reference. . . . .	23
5.1	Particle distribution at different time instants for the periodic array of cylinders. The color scale represents the magnitude of the velocity, marker particles are colored black. . . . .	26
5.2	Drag coefficient . . . . .	27

5.3	Steady state particle distribution, colored by velocity magnitude, for the periodic lattice of cylinders. Superimposed are the contour lines of the velocity magnitude. Red lines represent FEM solution from Morris et al. [26]. Black lines represent the SPH solution. Velocities are scaled by a factor $u_{\text{ref}} = 10^{-4}$	28
5.4	Measured streamwise velocity, $u_x$ , at probes placed at vertical lines $x = 0.5L$ (path 1), and $x = L$ (path 2). Only one resolution is shown per half path due to symmetry. Solid lines represent FEM solution.	28
5.5	Lid-driven cavity problem, marker particle setup, with normal vectors displayed.	29
5.6	Lid driven cavity $Re = 100$	30
5.7	Lid driven cavity $Re = 1000$	30
5.8	Lid driven cavity $Re = 10000$	30
5.9	Moving square inside a rectangular box. Problem setup and time law of the motion.	31
5.10	Drag coefficient for the moving square inside a rectangular box, for the three considered Reynolds numbers.	32
5.11	Vorticity and velocity magnitude contours at $t = 5$ for the moving square inside a rectangular box, for the three considered Reynolds numbers.	33
5.12	Vorticity and velocity magnitude contours at $t = 8$ for the moving square inside a rectangular box, for the three considered Reynolds numbers.	33
5.13	Vorticity field for the elliptical cylinder at $Re = 100$ and $Re = 1000$ .	35
5.14	Drag and lift coefficients as function of time for the elliptical cylinder at the $N = 50$ resolution.	35
5.15	Vorticity field for the triangular cylinder at $Re = 100$ and $Re = 1000$ . Snapshots at maximum lift.	36
5.16	Drag and lift coefficients as function of time for the triangular cylinder at the $N = 50$ resolution.	36
5.17	Sphere problem setup.	38
5.18	Drag coefficient of the sphere as function of the Reynolds number.	39
5.19	$u$ velocity around the sphere. Plane $y = 5$ (left) and plane $x = 10$ (right).	39
5.20	$v$ velocity around the sphere. Plane $z = 5$ (left) and plane $x = 9.5$ (right).	39
5.21	$w$ velocity around the sphere. Plane $y = 5$ (left) and plane $x = 9.5$ (right).	40
5.22	Dam break with adjacent hydrostatic tank. Interaction limiting of fluid particles disabled. Particles colored according to $v$ velocity component.	40
5.23	Dam break with adjacent hydrostatic tank. Interaction limiting of fluid particles enabled. Particles colored according to $v$ velocity component.	41
5.24	Geometry of the problem.	43
5.25	Discretization of the solid walls, using a cartesian lattice (top) and the new marker particle method (bottom).	43
5.29	Mixing of particles in the flow driven by the rotation of the polar rose shaped solids. Particles colored according to their tag value. Time evolution of the flow.	44

# List of Tables

5.1	Steady state average drag coefficient for the periodic array of cylinders. . . . .	27
5.2	Parameters of the gaussian acceleration function for the moving square. . . . .	31
5.3	Drag and lift coefficients, steady-state averaged values for the elliptical cylinder at $Re = 100$ . . . . .	35
5.4	Drag and lift coefficients, average and amplitude, for the elliptical cylinder at $Re = 1000$ . . . . .	36
5.5	Drag and lift coefficients for the triangular cylinder at $Re = 100$ . . . . .	37
5.6	Drag and lift coefficients for the triangular cylinder at $Re = 1000$ . . . . .	37



# Acronyms

<b>SPH</b>	Smoothed Particle Hydrodynamics
<b>ISPH</b>	Incompressible Smoothed Particle Hydrodynamics
<b>WCSPH</b>	Weakly Compressible Smoothed Particle Hydrodynamics
<b>LES</b>	Large Eddy Simulation
<b>CFD</b>	Computational Fluid Dynamics
<b>FGP</b>	Fixed Ghost Particles
<b>CAD</b>	Computer Aided Design



# Chapter 1

## Introduction

Smoothed Particle Hydrodynamics (SPH) is a mesh-free, Lagrangian particle-based method introduced independently by Gingold and Monaghan [11] and Lucy [19] in 1977. Initially developed for simulating astrophysical problems, it has since been extended to other applications, with Computational Fluid Dynamics (CFD) being of particular interest for this work. The method is formulated in terms of Lagrangian particles that carry the variables of interest for the system. The variation of any quantity can be determined through interactions between nearby particles, via a kernel function. The particles move with the flow, and thus advection is treated exactly. This makes the method especially well-suited for simulating flows with free surfaces and multiple phases, which among other applications, makes it a popular choice for industrial applications.

A particularly challenging aspect of the SPH framework is the difficulty in enforcing boundary conditions at solid boundaries, in contrast to grid-based methods, where this task is more straightforward. The task is complex since fluid particles close to a solid surface suffer from kernel truncation, caused by part of the kernel function's support being inside the solid domain. If not addressed properly, this can lead to the failure of the SPH approximation and produce unphysical results.

Over the years, various strategies have been developed to address boundary conditions in SPH, each with its own strengths and limitations. Early implementations, such as Monaghan's approach [24], placed solid particles along solid contours and applied artificial Lennard-Jones-like repulsive forces to fluid particles. This relatively simple method prevents particle penetration into the solid domain and can enforce no-slip conditions by adding a force term that mimics friction with the solid body. However, it requires calibration of the model parameters and introduces numerical errors due to kernel truncation.

Other single-layer methods include the normal-flux method of De Leffe et al.[6]. This method accounts for the missing kernel support of fluid particles near walls by evaluating the non-vanishing surface integral. Their work shows the applicability to flat wall scenarios, but does not address the applicability to complex geometries. Other approaches, like the one by Ferrand et al.[9] renormalize the kernel and kernel gradient summation with respect to the missing support near solid boundaries. This approach has the disadvantage of requiring increased computation due to the renormalization at every time step.

Alternative methods focus on enforcing the solid boundary conditions by filling the solid interior with particles. For instance, the so-called dummy particle methods fill the solid domain with fixed particles placed on a Cartesian grid. These particles are used to fill the support of nearby fluid particles, avoid kernel truncation, and improve the accuracy of the SPH approximation. The original approaches set the velocity field of solid particles to zero and computed density via summation from nearby fluid particles or through extrapolation from the fluid domain [32][16]. More recent approaches, such as the one by Adami et al. [1], extrapolate the pressure and velocity fields from nearby fluid particles to assign velocities to solid particles and enforce no-slip conditions. These methods are robust but suffer from the drawback of not representing the solid boundary geometry exactly, which can lead to numerical artifacts in complex geometries at low resolutions, since the solid boundary is represented by the nodes of a discrete Cartesian grid that may not accurately capture the actual geometry.

Another line of work developed around the idea of ghost particles [17], where ghost particles are placed inside the solid by mirroring the fluid particles at each time step. The ghost particles are assigned pressure and velocity fields, obtained from the fluid particles using various mirroring techniques. This

method can provide high accuracy when appropriate field mirroring techniques are used. In the work of Macia et al. [20], it is shown that the mirroring by Takeda et al. [33] outperforms other techniques. The main disadvantage of the ghost particle method is that mirroring particle positions becomes difficult for non-flat solid geometries, and it adds the computational cost of calculating the ghost particle positions at each time step. To address this issue, Marrone et al. [21] introduced the concept of Fixed Ghost Particles (FGP), where ghost particles are placed at fixed positions inside the solid, and each fixed ghost particle corresponds to a fixed interpolation point inside the fluid domain. The fluid interpolated fields are then mirrored to the FGP. This approach avoids the need to place ghost particles at each time step, but still faces challenges in placing interpolation nodes around sharp corners of the solid boundary.

To address issues with interpolation nodes without compromising accuracy, Marrone et al. [22] and later Antuono et al. [3] proposed local mirroring of fluid quantities. This technique mirrors fluid quantities pairwise between fluid and ghost particles, without requiring interpolation or averaging over a group of particles.

For any of the above approaches, the placement of particles inside the solid domain is not always straightforward. Consider a very convoluted solid boundary with sharp corners and regions of high curvature. To accurately represent the boundary, a high resolution is required to fit enough particles inside the solid to capture its geometric features. Furthermore, a seemingly simple but very thin rectangular solid is also problematic. Although it is easy to fill with particles, the thin walls of the solid can cause kernel truncation if not enough solid particles are placed inside it to fill the kernel radius. If the kernel radius extends beyond the solid, the no-slip condition will not be correctly enforced, and fluid particles from one side of the solid will interact with those on the other side, leading to unphysical results.

To address these issues, we propose a method based on the ideas of dummy and fixed ghost particles of solving the kernel truncation problem by filling the solid interior with particles, but that uses a single layer representation of the solid. The method condenses the interaction between fluid and solid particles to a single interaction with marker particles placed along the solid boundary. The placement of the particles along the solid boundary allows for a geometrically accurate resolution-independent representation of the solid geometry. The mirroring of the fluid phase to the solid particles will be done using the Takeda mirroring technique for high accuracy. This approach allows flexibility in working with any point discretization of the solid interface that can be generated by Computer Aided Design (CAD) softwares, admitting variable solid particle spacing. Additionally, condensing the solid representation to a single set of particles placed along the boundary allows for the use of techniques to handle sharp corners and thin obstacles.

## 1.1 Introduction to Smoothed Particle Hydrodynamics

In the SPH method, the fluid is discretized in a set of particles that move with the flow, and carry mass, density, pressure, velocity and potentially other quantities. To solve the equations of fluid mechanics numerically, one requires a discrete approximation of the spatial derivatives that appear in said equations. The SPH method provides such an approximation by interpolation from nearby particles through a kernel function.

To see how this interpolation is possible, we first represent a function  $f(\mathbf{r})$  as a convolution of itself with the Dirac delta function,  $\delta(\mathbf{r})$ ,

$$f(\mathbf{r}) = \int f(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')d\mathbf{r}'. \quad (1.1)$$

The next step is to substitute the delta function with a function that approximates it. Such function is known as the kernel function,  $W(\mathbf{r}, h)$ , where  $h$  is a parameter, the smoothing length,

$$f(\mathbf{r}) \approx \int f(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}'. \quad (1.2)$$

Using integration by parts, we also find an expression for the gradient of  $f$ ,

$$\nabla f(\mathbf{r}) \approx - \int f(\mathbf{r}')\nabla W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}'. \quad (1.3)$$

Now we approximate the infinite sum, as a discrete Riemann sum, over the values of the function at the particle positions,  $f_b$ , and volume elements represented by the particles, that satisfy  $m_b = \rho_b V_b$ . We therefore have

$$\begin{aligned} f(\mathbf{r}_a) &\approx \sum_b V_b f_b W(\mathbf{r}_a - \mathbf{r}_b, h) = \sum_b \frac{m_b}{\rho_b} f_b W(\mathbf{r}_a - \mathbf{r}_b, h), \\ \nabla f(\mathbf{r}_a) &\approx \sum_b V_b f_b \nabla W(\mathbf{r}_a - \mathbf{r}_b, h) = \sum_b \frac{m_b}{\rho_b} f_b \nabla W(\mathbf{r}_a - \mathbf{r}_b, h). \end{aligned} \quad (1.4)$$

Where we have evaluated the left hand side at the position of particle  $a$ , as we will do in SPH computations. However the previous gradient approximation is rarely applied to solving the equations of fluid mechanics. Instead, alternative expressions are derived, that have more desirable properties, such as anti-symmetry between particles  $a$  and  $b$  that allows for conservative solutions to the equations of motion. So in practical applications, it is common to see the differential operators discretized in other forms [34][30][23].

The choice of a kernel function is crucial for the accuracy of the SPH approximation. A good kernel function must satisfy some properties.

- Convergence to the delta function for  $h \rightarrow 0$ ,

$$\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r}). \quad (1.5)$$

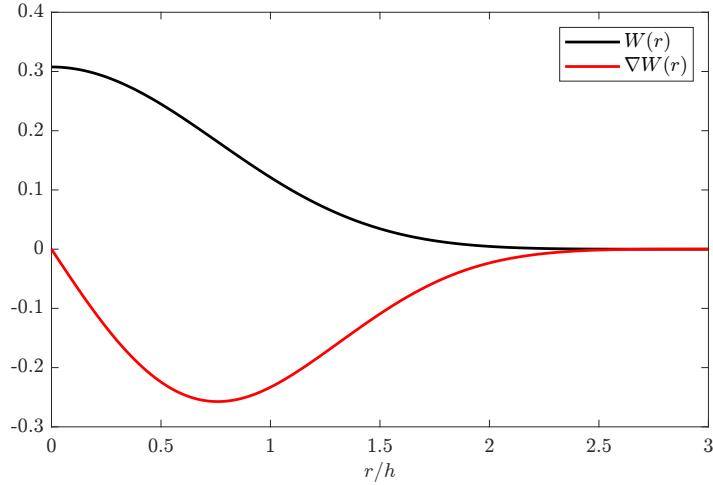
- Normalization,

$$\int W(\mathbf{r}, h) d\mathbf{r} = 1. \quad (1.6)$$

- Symmetry,

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h). \quad (1.7)$$

Moreover, for efficiency reasons it is desirable that the kernel function has compact support, i.e. that it vanishes outside a certain threshold radius,  $r_c$ , so the interpolation is limited to near neighbors.



**Figure 1.1:** Quintic spline kernel, and kernel gradient.

In particular in this work we use the quintic spline kernel, with  $r_c = 3h$ , which is a popular choice in SPH simulations, see Morris et al.[26], and has the following expression

$$W(r, h) = K_d \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5 & \text{if } 0 \leq q < 1, \\ (3-q)^5 - 6(2-q)^5 & \text{if } 1 \leq q < 2, \\ (3-q)^5 & \text{if } 2 \leq q < 3, \\ 0 & \text{if } q \geq 3, \end{cases} \quad (1.8)$$

Where  $q = \frac{r}{h}$ , and  $K_d$  is a normalization constant that depends on the dimension of the problem. In two and three dimensions

$$K_2 = \frac{7}{478\pi h^2}, \quad K_3 = \frac{1}{120\pi h^3}. \quad (1.9)$$

And its gradient is given by

$$\nabla W(\mathbf{r}, h) = -\hat{\mathbf{r}} |F(r)| = \hat{\mathbf{r}} \frac{-5K_d}{h} \begin{cases} (3-q)^4 - 6(2-q)^4 + 15(1-q)^4 & \text{if } 0 \leq q < 1, \\ (3-q)^4 - 6(2-q)^4 & \text{if } 1 \leq q < 2, \\ (3-q)^4 & \text{if } 2 \leq q < 3, \\ 0 & \text{if } q \geq 3, \end{cases} \quad (1.10)$$

Where  $\hat{\mathbf{r}}$  is the unit vector pointing in the direction of  $\mathbf{r}$ , and  $|F(r)| \geq 0$  is the magnitude of the gradient. Another important parameter choice is the ratio  $\frac{h}{\Delta x}$ , which determines the number of neighbors each particle has. In this work we set  $\frac{h}{\Delta x} = 1$ .

# Chapter 2

## Method

### 2.1 Governing Equations

The governing equations of fluid mechanics, for an isothermal fluid, are the continuity and the Navier-Stokes equations. Written in a Lagrangian frame of reference, these are given by

$$\begin{aligned}\frac{d\rho}{dt} &= -\rho \nabla \cdot \mathbf{u} \\ \rho \frac{d\mathbf{u}}{dt} &= -\nabla p + \eta \nabla^2 \mathbf{u} + \mathbf{g}.\end{aligned}\tag{2.1}$$

Where  $\rho$  is the fluid density,  $\mathbf{u}$  is the fluid velocity,  $p$  is the pressure,  $\eta$  is the dynamic viscosity, and  $\mathbf{g}$  a body force, usually gravity.

We are interested in the case of an incompressible fluid, to model incompressible flows in SPH, two approaches are commonly used, a true incompressible formulation, ISPH [14]. And a weakly compressible method WCSPH[26][24]. In the former method, the pressure needs to be solved for at every time step to enforce the velocity field to be divergence-free, which can be computationally expensive. In this work we will follow the Weakly Compressible Smoothed Particle Hydrodynamics (WCSPH) approach, where the density is allowed to fluctuate around a reference value, and the pressure is computed from the density using an equation of state. We use an equation of state of the form

$$p(\rho) = p_0 \left( \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right) + \chi.\tag{2.2}$$

Where  $p_0$  and  $\rho_0$  are the reference pressure and density respectively and  $\chi$  is the background pressure, in this work we will set  $\chi = 0$ . The reference pressure is set through an artificial speed of sound  $c$  as

$$p_0 = \frac{\rho_0 c^2}{\gamma}.\tag{2.3}$$

The larger the speed of sound, the closer the weakly compressible fluid is to an incompressible one. But with higher speed of sound, comes the restriction of a smaller time step, thus increasing the computational cost. It is a common practice to set the speed of sound high enough to limit the density fluctuations to 1% of the reference density. In Morris, et al. [26] it is shown that the relation between speed of sound and allowed density fluctuation,  $\delta = \frac{\Delta\rho}{\rho_0}$ , is

$$c \sim \max \left( \frac{U_{\max}}{\sqrt{\delta}}, \sqrt{\frac{\nu U_{\max}}{L_0 \delta}}, \sqrt{\frac{F L_0}{\delta}} \right) \xrightarrow{\delta=0.01} c \sim \max \left( 10 U_{\max}, 10 \sqrt{\frac{\nu U_{\max}}{L_0}}, 10 \sqrt{\frac{F}{L_0}} \right)\tag{2.4}$$

### 2.2 Numerical Method

When formulating the equations of fluid motion in the SPH method, we get a set of differential equations for the particle density and velocity. This equations are a result applying the spatial

derivatives from Eq.(2.1) for the SPH interpolants of the fields. These interpolants can be differentiated exactly, and produce discrete expressions in terms of summations over the neighboring particles. However there are multiple ways to write the derivatives of the interpolants, each with different accuracy and conservation properties, see [23].

Additionally the WSCPH method suffers from the well known tensile instability[25]. This instability is caused by particles with negative pressures being attracted and accumulating in regions of the flow, and eventually creating void regions in the flow, thus making the solution unphysical. Multiple techniques are used to get rid of this instability, a common practice is to add a constant background pressure in order to make the pressure field positive in the whole domain, see [22]. This approach introduces additional numerical dissipation, that can be of the order of the physical viscosity for some problems, thus modifying the results. In this work we will follow the transport velocity formulation of Adami et al. [2], in which the particles are advected with a transport velocity,  $\tilde{\mathbf{u}}$ , different of their momentum velocity,  $\mathbf{u}$ , homogenizing the particle configuration without introducing numerical dissipation.

### 2.2.1 Density

There are two main options to compute the density of a particle, the first one is to integrate in time the discrete version of the continuity equation, this approach is suitable for free surface flows, however it introduces more errors in the density field over time. The second option is to compute the density directly from a summation of the kernel function over the neighboring particles. This means we have an explicit equation for the density that only depends on the particle configuration. This option is more stable, however it is more computationally expensive, and it can not be used straightforwardly in free surface flows. In general in this work we will use the summation form of the density equation, and resort to the continuity equation when needed.

#### Continuity Equation

As in [1] this work we follow one of the formulations of [23] for the continuity equation, that also works for multi-phase scenarios. The continuity equation from Eq.(2.1) is discretized as

$$\frac{d\rho_a}{dt} = \rho_a \sum_b \frac{m_b}{\rho_b} \mathbf{u}_{ab} \cdot \nabla_a W_{ab}. \quad (2.5)$$

Where  $\mathbf{u}_{ab} = \mathbf{u}_a - \mathbf{u}_b$  is the relative velocity between particles  $a$  and  $b$ , and  $\nabla_a W_{ab} = \nabla_a W(\mathbf{r}_a - \mathbf{r}_b, h)$  is the kernel gradient with respect to the coordinates of particle  $a$ . The summation is over all particles  $b$  that are within the support of the kernel function.

#### Density Summation

In the case of using the density summation equation, the density of a particle  $a$  is computed directly from the positions of other particles, as,

$$\rho_a = m_a \sum_b W_{ab}. \quad (2.6)$$

This expression increases the density in regions where the particles are more tightly packed, and decreases it in regions where the particles are more sparse. However it is more computationally expensive to compute the density in this way, since it requires a second double loop over all particles and all neighbors, unlike the continuity equation that can be solved in the same loop as the force computation.

In particular, Eq.2.6 takes the form presented in Hu and Adams [13], which still holds for multi-phase problems, and conserves mass exactly.

### 2.2.2 Momentum Equation

The discretization of the momentum equation follows that of [13], which allows multi-phase problems, and includes the transport velocity formulation from Adami et al. [2], and reads as

$$\frac{d\mathbf{u}_a}{dt} = \frac{1}{m_a} \sum_b (V_a^2 + V_b^2) \left[ -\tilde{p}_{ab} \nabla_a W_{ab} + \tilde{\eta}_{ab} \frac{\mathbf{u}_{ab}}{r_{ab}} \frac{\partial W}{\partial r_{ab}} + \frac{1}{2} (\mathbf{A}_a + \mathbf{A}_b) \nabla_a W_{ab} \right] + \mathbf{g}. \quad (2.7)$$

Where  $V_a = \frac{m_a}{\rho_a}$  and  $V_b = \frac{m_b}{\rho_b}$  are the volumes of particles  $a$  and  $b$ . The three terms inside square brackets correspond to the discretization of the pressure gradient, the viscous force and an additional term that is a result of the modified advection velocity.

The inter particle averaged viscosity, and density weighted pressure are

$$\tilde{p}_{ab} = \frac{\rho_b p_a + \rho_a p_b}{\rho_a + \rho_b}, \quad (2.8)$$

and

$$\tilde{\eta}_{ab} = \frac{2\eta_a \eta_b}{\eta_a + \eta_b} \quad (2.9)$$

respectively. The third term accounts for the divergence of the tensor  $A$  which is defined as

$$\mathbf{A} = \rho \mathbf{u} \otimes (\tilde{\mathbf{u}} - \mathbf{u}). \quad (2.10)$$

The transport velocity is computed from the momentum velocity plus the gradient of a constant background pressure  $p_B$ , as

$$\tilde{\mathbf{u}}_a(t + \delta t) = \mathbf{u}_a + \delta t \left( \frac{d\mathbf{u}_a}{dt} - \frac{p_B}{m_a} \sum_b (V_a^2 + V_b^2) \nabla_a W_{ab} \right) = \mathbf{u}_a + \delta t \left( \frac{d\mathbf{u}_a}{dt} + \mathbf{F}_{p_B} \right). \quad (2.11)$$

Where we write  $\mathbf{F}_{p_B}$  as this gradient for brevity.

This pressure gradient vanishes in regions where the particle distribution is uniform, and in regions where voids would be created, it corrects the trajectory of particles, thus avoiding the tensile instability. And by not including the gradient of the background pressure in the momentum equation, no numerical dissipation is introduced. The magnitude of  $p_B$  can be adjusted, we take it to be of the order of the reference pressure  $p_0$ . Also notice how this term can be computed together with the force computation, thus not increasing the computational cost of the method.

### 2.3 Time Stepping

To integrate the system in time we use the kick-drift-kick scheme, where the velocity is updated in two half steps, and the position is updated in the middle of the two velocity updates. When using the density summation equation, the scheme is as follows

$$\begin{aligned} \mathbf{u}^{n+\frac{1}{2}} &= \mathbf{u}^n + \frac{\delta t}{2} \frac{d\mathbf{u}}{dt}^{n-\frac{1}{2}}, \\ \tilde{\mathbf{u}}^{n+\frac{1}{2}} &= \mathbf{u}^{n+\frac{1}{2}} + \frac{\delta t}{2} \mathbf{F}_{p_B}, \\ \mathbf{r}^{n+1} &= \mathbf{r}^n + \delta t \tilde{\mathbf{u}}^{n+\frac{1}{2}}, \\ \rho^n &= m_a \sum_b W_{ab} \\ \mathbf{u}^{n+1} &= \mathbf{u}^{n+\frac{1}{2}} + \frac{\delta t}{2} \frac{d\mathbf{u}}{dt}^{n+\frac{1}{2}}. \end{aligned} \quad (2.12)$$

When computing the density from the continuity equation, the scheme is the same, but the density is updated in the same way as the velocity, in two half steps.

$$\begin{aligned}
\mathbf{u}^{n+\frac{1}{2}} &= \mathbf{u}^n + \frac{\delta t}{2} \frac{d\mathbf{u}}{dt}^{n-\frac{1}{2}}, \\
\tilde{\mathbf{u}}^{n+\frac{1}{2}} &= \mathbf{u}^{n+\frac{1}{2}} + \frac{\delta t}{2} \mathbf{F}_{p_B}, \\
\rho^{n+\frac{1}{2}} &= \rho^n + \frac{\delta t}{2} \frac{d\rho}{dt}^{n-\frac{1}{2}}, \\
\mathbf{r}^{n+1} &= \mathbf{r}^n + \delta t \tilde{\mathbf{u}}^{n+\frac{1}{2}}, \\
\mathbf{u}^{n+1} &= \mathbf{u}^{n+\frac{1}{2}} + \frac{\delta t}{2} \frac{d\mathbf{u}}{dt}^{n+\frac{1}{2}} \\
\rho^{n+1} &= \rho^{n+\frac{1}{2}} + \frac{\delta t}{2} \frac{d\rho}{dt}^{n+\frac{1}{2}}.
\end{aligned} \tag{2.13}$$

The pressure is always updated simultaneously with the density, using the equation of state. To ensure stability of the method, the time step is set to the minimum of three conditions, a CFL like condition, a viscous condition and a body force condition.

$$\begin{aligned}
\delta t &\leq 0.25 \frac{h}{c + |\mathbf{u}|_{max}}, \\
\delta t &\leq 0.125 \frac{h^2}{\nu}, \\
\delta t &\leq 0.25 \sqrt{\frac{h}{|\mathbf{g}|}}.
\end{aligned} \tag{2.14}$$

## Chapter 3

# Solid Wall Modelling

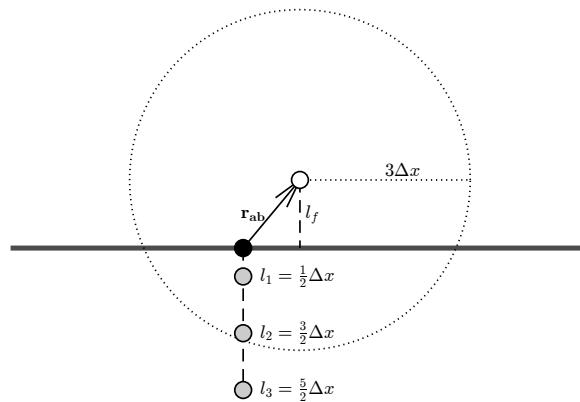
The central topic of this work is the formulation, implementation, and validation of a new modeling technique for the no-slip condition on solid walls in SPH. This technique aims to improve the particle-based representation of complex wall geometries.

### 3.1 Discretization of the Solid Interface

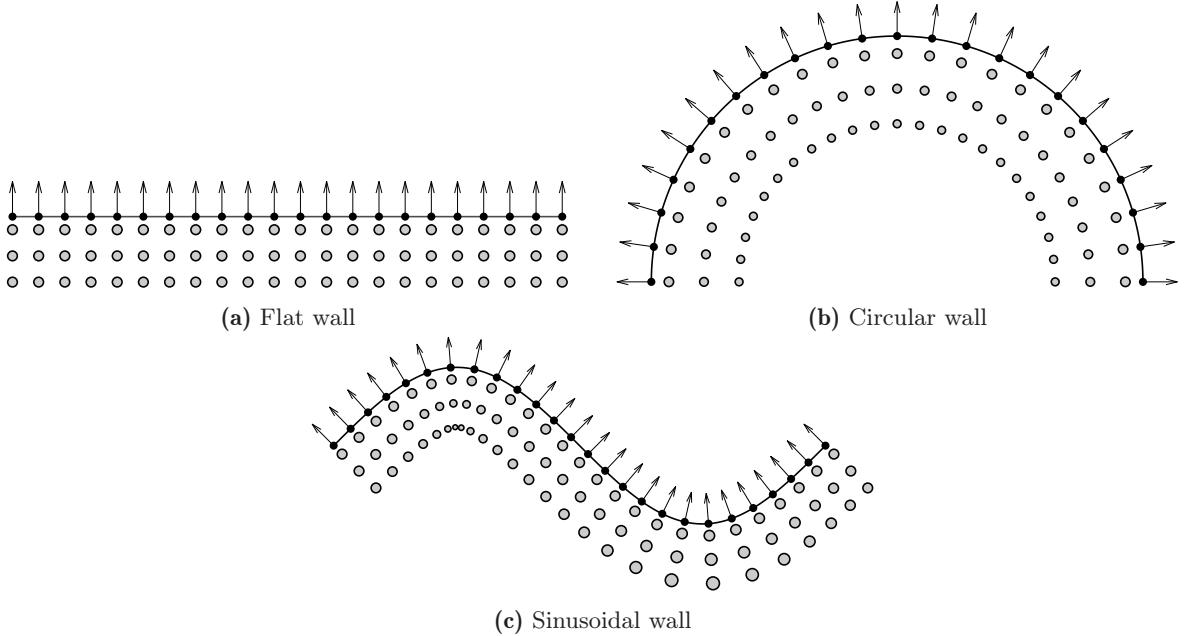
The main idea is to discretize the fluid-solid interface curve (or surface in 3D), by placing a set of marker particles along the solid wall. Each of these particles is associated with a length element  $\Delta s$  (in 2D) or an area element (in 3D), a unit normal vector  $\hat{\mathbf{n}}$  pointing outward from the solid, and a scalar curvature  $\kappa$ . The length/area element represents the piece of curve/surface that belongs to the marker particle when the solid interface is discretized and need not be the same for all of them.

Each marker particle generates three virtual particles inside the solid. These particles are aligned opposite to the normal vector and spaced at fixed distances of  $l_1 = \frac{h}{2}$ ,  $l_2 = \frac{3h}{2}$ , and  $l_3 = \frac{5h}{2}$  from the marker particle, see Fig. 3.1. This spacing is specifically chosen to ensure complete coverage of the quintic kernel's support radius,  $r_c = 3h$ , for a fluid particle near the wall, similar to the approach of Adami et al. [1]. Other kernels might need different number of virtual particles and different spacings to fill the kernel support.

Marker particles are initialized with a normal vector and the volumes of their associated virtual particles. During the simulation, fluid particles interact with marker particles whenever they fall within the marker particle's kernel support. When such interactions occur, the mirrored fields for the virtual particles are calculated. The resulting force and density contributions are then algebraically computed as if actual particles were positioned at the virtual particles' locations. This method is particularly well-suited for the mirroring procedure described in the following sections. Notably, the



**Figure 3.1:** Schematic representation of the virtual particles (grey) generated by a marker particle (black) interacting with a fluid particle (white). Relevant distances for the interaction are shown.



**Figure 3.2:** Different wall geometries discretized with marker particles (black). The virtual particles (grey) are generated in the opposite direction of the normal vector. Note that the curvature of the wall affects the distribution and volume of the virtual particles, as seen in the spacing and size differences between flat, circular, and sinusoidal geometries.

virtual particles inherit the normal vector directly from their marker particle, and their positions, which are fixed, are straightforward to compute based on the marker particle's position.

## 3.2 Volume of Virtual Particles

With this arrangement of the marker particles, the distribution of virtual particles generated by marker particles is not Cartesian, but depends on the curvature of the wall. Therefore their volume is a function of the curvature wall. This is a key feature of the method, as it allows the virtual particles to adapt to the curvature of the wall.

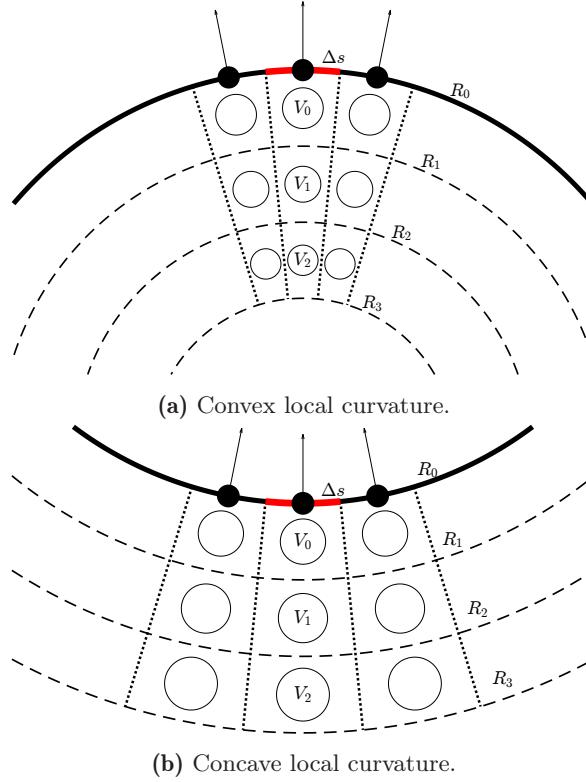
To derive the volume of virtual particles, we first examine how particle volume is typically assigned in SPH. In 2D Cartesian lattices, particles are typically assigned a volume  $\Delta x^2$ , corresponding to their cell size. Each particle occupies a cell and is therefore assigned a volume of  $\Delta x^2$ . This volume weights the particle's contribution to the summations in the SPH equations. With a non Cartesian particle distribution, the volume of the particles is no longer constant.

When the solid interface is discretized with marker particles, each is assigned a curvature  $\kappa$ , equal to the wall's curvature at its position. The curvature is a scalar with dimensions of  $1/L$ , and it is positive for convex walls, and negative for concave walls. For an arbitrary wall, we approximate the interface locally in the neighborhood of a particle using a circle with the same curvature. This circle, known as the osculating circle, has a radius  $R_0 = \frac{1}{|\kappa|}$ .

Consider the case in which the interface is convex, i.e.  $\kappa > 0$ , seen in Fig. 3.3a. We label the virtual particles as 1, 2, 3 with 1 being the closest to the wall. To each marker particle belongs a circular sector, of arc length  $\Delta s$ . We imagine concentric circles that cross the midpoint between virtual particles. Since the spacing between virtual particles is  $\Delta x$ , the radius of the circles is given by

$$R_n = R_0 - n\Delta x \text{ for } n = 1, 2, 3, \quad (3.1)$$

Where  $n$  is the layer index of the virtual particles. These circles confine the virtual particles inside cells, limited by the concentric circles on the top and bottom, by the limits of the circular sector on the left and right. To determine the volume of the virtual particles, we must determine the volume



**Figure 3.3:** Volume of the virtual particles in a convex and concave scenario. Marker particles with their normal vectors represented in black. Length element  $\Delta s$  highlighted in red. Circles dividing the virtual particles in layers, with radius  $R_0$  dashed lines. It can be seen that for convex curvature the volume of the first layer of particles is larger than the second and third. The opposite is true for concave curvature.

of the cells. This is given by the area of each annular region between circles, times the fraction of the full circle that the circular sector represents.

The fraction of the circular sector can be computed through the fraction of the perimeter it represents. The total osculating circle perimeter is  $2\pi R_0$ , and the arc length of our sector is  $\Delta s$ . So the sector of interest represents a fraction of the circle given by

$$\frac{\Delta s}{2\pi R_0} = \frac{|\kappa| \Delta s}{2\pi} \quad (3.2)$$

And the area of each annular region is just given by the difference of the area of the outer circle minus the area of the inner circle

$$\pi R_{n-1}^2 - \pi R_n^2 = \pi (2R_0 \Delta x + (1 - 2n) \Delta x^2). \quad (3.3)$$

Multiplying with Eq.3.2 we get the volume of the virtual particles.

$$V_n = \frac{1}{2} (2\Delta x + (1 - 2n) \Delta x^2 |\kappa|) \Delta s \quad (3.4)$$

In the concave case, the radius of the circles grows towards the inside of the wall, and it is given by

$$R_n = R_0 + n\Delta x \text{ for } n = 1, 2, 3. \quad (3.5)$$

Therefore the volume is now

$$V_n = \pi (R_n^2 - R_{n-1}^2) \frac{|\kappa| \Delta s}{2\pi} = \frac{1}{2} (2\Delta x - (1 - 2n) \Delta x^2 |\kappa|) \Delta s. \quad (3.6)$$

If we consider the curvature with sign, positive for convex, negative for concave, we can write both cases in a single expression as

$$V_n = \frac{1}{2} (2\Delta x + (1 - 2n) \Delta x^2 \kappa) \Delta s. \quad (3.7)$$

We can see that in the particular case of a flat wall,  $\kappa = 0$ , and with marker particles spaced with  $\Delta s = \Delta x$ , we recover the volume of a flat particle, as expected,

$$V_n^{\text{flat}} = \frac{1}{2} (2\Delta x) \Delta x = \Delta x^2. \quad (3.8)$$

Additionally, the volume expression has the intuitively expected properties that in the convex case  $V_1 > V_2 > V_3$ , and in the concave case  $V_1 < V_2 < V_3$ .

Similarly, for 3D surfaces, the derivation follows similar steps. Here, curvature is treated as a scalar with dimensions of  $1/L$ . We now locally approximate the surface by a sphere with radius  $R_0 = \frac{1}{|\kappa|}$ . We define three additional spheres, with radius  $R_n = R_0 \pm n\Delta x$  for  $n = 1, 2, 3$ , with the plus sign for the concave case, and the minus sign for the convex case. The virtual particles are contained between consecutive spheres. Similarly to the 2D case, the volume of the virtual particles is given by the difference of the volumes between consecutive spheres, times the fraction of the total sphere, the sector represents.

In 3D the fraction can be computed as the area of the marker particle, divided by the total surface area of the sphere, this fraction is

$$\frac{\Delta s}{4\pi R_0^2} = \frac{|\kappa|\Delta s}{4\pi}. \quad (3.9)$$

And the volume in between spheres for the convex case, (similar steps can be followed for the concave case)

$$V_n = \frac{4}{3}\pi R_{n-1}^3 - \frac{4}{3}\pi R_n^3 = \frac{4}{3}\pi (3R_0^2\Delta x + R_0\Delta x^2(3 - 6n) + \Delta x^3(3n^2 - 3n + 1)) \quad (3.10)$$

Multiplying this times the fraction of the sphere we get the volume of the virtual particles

$$V_n = \frac{1}{3} (3\Delta x + \kappa\Delta x^2(3 - 6n) + \kappa^2\Delta x^3(3n^2 - 3n + 1)) \Delta s \quad (3.11)$$

This expression retains the same nice properties that the 2D expression has. This properties are that in the flat case, that is,  $\kappa = 0$ ,  $\Delta s = \Delta x^2$ , we recover the volume of a flat particle,  $\Delta x^3$ . And in the convex case,  $V_1 > V_2 > V_3$ , in the concave case  $V_1 < V_2 < V_3$ .

### 3.3 Velocity and Pressure Mirroring

The solid wall boundary condition is enforced by locally mirroring the velocity and pressure from the fluid phase onto the virtual particles. Here, "locally" implies that each fluid particle generates mirrored fields in the virtual particles it interacts with. This method employs the velocity mirroring approach introduced by Takeda et al. [33], which has also been used by other researchers, such as in [3] and [22]. The technique relies on linear extrapolation, it assigns a mirrored velocity to the virtual particles based on the fluid particle's velocity and the wall's velocity.

De Leffe et al. [7] highlighted that instabilities can arise from unphysical energy exchanges between the fluid and solid. To solve this issue, their work proposed distinct mirroring strategies for different terms in the governing equations. Specifically, the hyperbolic terms require free-slip mirroring, while the viscous terms demand no-slip mirroring. In practice, this approach involves constructing two separate velocity fields for the virtual particles: one that mirrors the normal component of the velocity for the continuity equation, and another that mirrors the tangential component for the viscous term. To implement no-slip conditions, these two velocity fields are computed separately for the continuity and viscous terms. To implement free-slip conditions, both terms use the same free-slip mirroring.

For a fluid particle  $a$  with velocity  $\mathbf{u}_a$ , interacting with a virtual particle  $b$  in the  $n$ -th layer, the mirrored velocities are

$$\mathbf{u}_b^{\text{visc}} \begin{cases} \mathbf{u}_b^{\text{visc}} \cdot \hat{\mathbf{n}} = \mathbf{u}_a \cdot \hat{\mathbf{n}} \\ \mathbf{u}_b^{\text{visc}} \cdot \hat{\boldsymbol{\tau}} = \left[ (\mathbf{u}_w - \mathbf{u}_a) \frac{l_n}{\max(l_f, 0.25\Delta x)} + \mathbf{u}_w \right] \cdot \hat{\boldsymbol{\tau}} \end{cases}, \quad (3.12)$$

$$\mathbf{u}_b^{\text{cont}} \begin{cases} \mathbf{u}_b^{\text{cont}} \cdot \hat{\mathbf{n}} = \left[ (\mathbf{u}_w - \mathbf{u}_a) \frac{l_n}{\max(l_f, 0.25\Delta x)} + \mathbf{u}_w \right] \cdot \hat{\mathbf{n}} \\ \mathbf{u}_b^{\text{cont}} \cdot \hat{\boldsymbol{\tau}} = \mathbf{u}_a \cdot \hat{\boldsymbol{\tau}} \end{cases}. \quad (3.13)$$

Where  $\hat{\mathbf{n}}$  and  $\hat{\boldsymbol{\tau}}$  are the unit normal and tangential vectors to the wall. The velocity of the wall is  $\mathbf{u}_w$ . The distance  $l_n$  is the distance from the  $n$ th virtual particle to the wall, which is a known constant. The distance  $l_f$  is the projection of the fluid particle's position vector onto the wall's normal.

$$l_f = \mathbf{r}_{ba} \cdot \hat{\mathbf{n}}. \quad (3.14)$$

We apply the maximum function to this distance to avoid division by zero, when the fluid particle is very close to the wall.

For the pressure field we use the approach from [3], where we substitute the values for the distances of the fluid and virtual particles. This results in a pressure for the virtual particle given by

$$p_b = p_a - \rho_a (l_f + l_{wn}) (\mathbf{g} - \mathbf{a}_w) \cdot \hat{\mathbf{n}}. \quad (3.15)$$

Where  $\mathbf{g}$  is the gravity vector,  $\mathbf{a}_w$  is the acceleration of the wall and  $p_a$  and  $\rho_a$  are the pressure and density of the fluid particle. Once the pressure is known, the density of the virtual particle is computed by inverting the equation of state, Eq.2.2,

$$\rho_b = \rho_0 \left[ \frac{p_b - \chi}{p_0} + 1 \right]^{\frac{1}{\gamma}}. \quad (3.16)$$

### 3.4 Force and Density Computation

When computing the force contribution of a virtual particle on a fluid particle, Eq.2.7 is no longer valid. Since it assumes the volumes of particles  $a$  and  $b$ , given by  $V_i = \frac{m_i}{\rho_i}$ , to be of the same magnitude up to the density fluctuations. This is no longer the case, since the resolution at which the solid interface is discretized,  $\Delta s$ , can be different from the resolution in the fluid phase,  $\Delta x$ . Thus possibly creating virtual particles with a widely different volume to the fluid phase. The overall volume covered by the virtual particles will be the same since for coarser resolutions there are less of them, and for finer resolutions there are more of them. But the volume of each individual virtual particle can be different. Therefore we must adopt a force formulation that is linear in volume and "one sided", only considering the volume of particle  $b$ . This is how more fundamental SPH formulations, treat the volume weighting of the particles, see Eq.1.4. We can do this without loss of generality for multiphase problems, since fluid-wall interactions happen on a particle by particle basis, the different pressures and densities of the interacting phase is already accounted for in Eq.3.15 and Eq.3.16.

We adopt a formulation similar to Eq.2.7, substituting the volume weighting factor

$$\frac{1}{m_a} (V_a^2 + V_b^2),$$

with a factor that only takes into account the volume of the virtual particle,

$$\frac{2V_b}{\rho_a}.$$

So the force contribution of a marker particle on a fluid particle  $a$  is

$$\frac{1}{\rho_a} \sum_{n=1}^3 2V_n \left[ -\tilde{p}_{an} \nabla_a W_{an} + \eta_a \frac{\mathbf{u}_{an}}{r_{an}} \frac{\partial W}{\partial r_{an}} + \frac{1}{2} \mathbf{A}_a \nabla_a W_{an} \right] + \mathbf{g}. \quad (3.17)$$

Where the summation is over the virtual particles of the marker particle. The density weighted pressure is  $\tilde{p}_{an} = \frac{\rho_n p_a + \rho_a p_n}{\rho_a + \rho_n}$ , where  $p_n$  and  $\rho_n$  are the pressure and density of the virtual particle, computed using Eq.3.15 and Eq.3.16. And for the viscosity term,  $\mathbf{u}_{an} = \mathbf{u}_a - \mathbf{u}_n^{\text{visc}}$  where  $\mathbf{u}_n^{\text{visc}}$  is the mirrored velocity, according to Eq.3.12.

Analogously, the contribution of the virtual particles to the gradient of the background pressure term is

$$\frac{p_B}{\rho_a} \sum_{n=1}^3 2V_n \nabla_a W_{an}. \quad (3.18)$$

The contribution to the continuity equation is

$$\frac{1}{\rho_a} \sum_{n=1}^3 V_n \mathbf{u}_{an} \cdot \nabla_a W_{an}. \quad (3.19)$$

Where  $\mathbf{u}_{an} = \mathbf{u}_a - \mathbf{u}_n^{\text{cont}}$  and  $\mathbf{u}_n^{\text{cont}}$  is the mirrored velocity, according to Eq.3.13.

And when using the density summation approach, the contribution to the summation is

$$\sum_{n=1}^3 \rho_0 V_n W_{an}. \quad (3.20)$$

### 3.5 Computing the Normal Vectors and Curvature

The method detailed in the previous sections requires the marker particles to be equipped with a normal vector oriented outwards from the solid, and a curvature value. This information is needed to compute the volume and positions of the virtual particles. In the following we detail the method we used to determine the normal vectors and curvatures. This is, however, not a key feature of the method, and other tools to compute the normal vectors and curvatures could be used.

Preliminary to the normal vector computation, we must give every marker particle a vector that points outside the solid, to be able to correctly orient the normal vector. We do this by computing a vector given by the kernel averaged distance vector to the fluid particles in the neighborhood of the marker particle. This vector is then normalized and set as an initial guess for the normal. In the case of a solid surface that is not in contact with the fluid, the initial guess for the normal vector has to be manually prescribed.

When it is possible to find this initial guess from the fluid particles, the approach is as follows. Let a marker particle  $i$  have a position  $\mathbf{r}_i$ , and a set of surrounding fluid particles  $j$  with positions  $\mathbf{r}_j$ . We compute the initial guess for the normal vector  $\hat{n}_i^0$  as

$$\hat{n}_i^0 = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \text{ with } \mathbf{n}_i = \sum_{j \in \text{fluid}} \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|} W_{ij}. \quad (3.21)$$

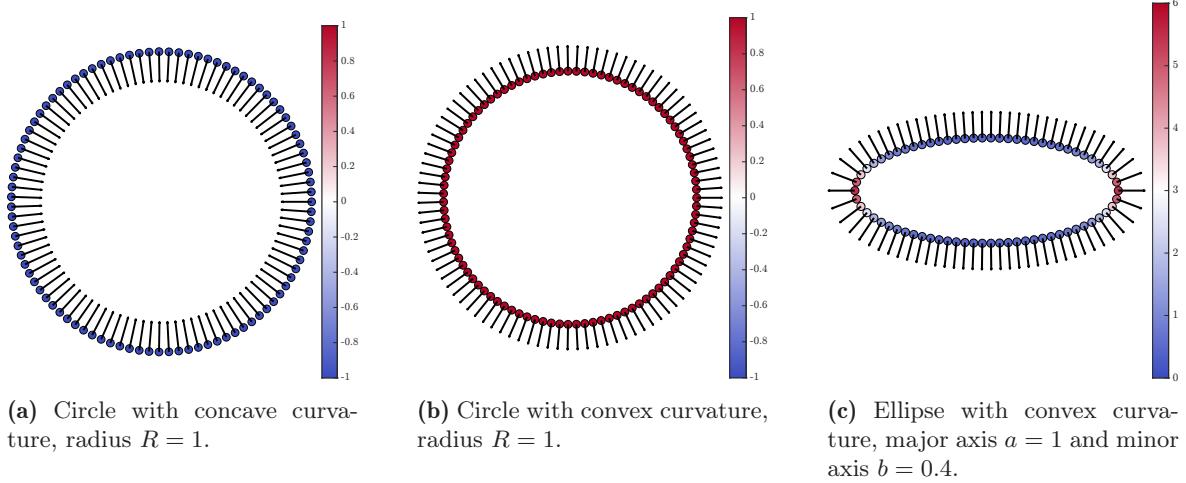
Once this initial guess is computed, we compute the normal vector  $\hat{n}_i$  by kernel averaging the vectors perpendicular to the distance vector to other marker particles inside the kernel support. In two dimensions, for an arbitrary vector  $\mathbf{v} = v_x \hat{x} + v_y \hat{y}$ , its perpendicular vector is  $\mathbf{v}^\perp = -v_y \hat{x} + v_x \hat{y}$ . In the averaging operation, we must make sure all perpendicular vectors are pointing outwards, to do so, we check the dot product with the initial guess, and if it is negative, we flip the vector. The normal vector is then computed as

$$\hat{n}_i = \sum_{j \in \text{marker}} \sigma_{ij} \frac{\mathbf{r}_{ij}^\perp}{\|\mathbf{r}_{ij}^\perp\|} W_{ij} \text{ with } \sigma_{ij} = \begin{cases} +1 & \text{if } \mathbf{r}_{ij}^\perp \cdot \hat{n}_i^0 > 0 \\ -1 & \text{if } \mathbf{r}_{ij}^\perp \cdot \hat{n}_i^0 < 0 \end{cases}. \quad (3.22)$$

This vector is then normalized to have unit length. Once the normal vectors are computed, the curvature is computed as the divergence of the normal vector,  $\kappa = \nabla \cdot \hat{n}$ . In the SPH framework, for a marker particle  $i$ , the curvature is computed as

$$\kappa_i = \sum_{j \in \text{marker}} \max \left( \kappa_{\max}, \frac{(\hat{n}_i - \hat{n}_j) \cdot \hat{e}_{ab}}{\|\mathbf{r}_{ij}\|} \right) W_{ij}. \quad (3.23)$$

Where  $\hat{e}_{ab} = \frac{\mathbf{r}_{ab}}{\|\mathbf{r}_{ab}\|}$  is the unit vector between particles  $a$  and  $b$ . And  $\kappa_{\max}$  is a parameter that limits the curvature contribution from any single particle, in this work is set as  $\kappa_{\max} = \frac{1}{2\Delta x}$ , the reasoning for this choice lies in the treatment of sharp corners, and it is detailed in the next section.



**Figure 3.4:** Examples of numerically computed normal vectors, and curvature for smooth geometries. The normal vectors are represented in black, and the curvature by the color scale.

In Fig.3.4 we present the normal vectors and curvature this method produces for various smooth geometries. In particular we present the case of two circles with radius  $R = 1$ , one with concave curvature, and the other with convex curvature. As can be seen the normal vectors are correctly computed and match the expected orientation. With respect to the curvature, we expect a constant value of  $\kappa = \mp \frac{1}{R}$ , respectively for the concave and convex cases, this value is computed exactly, up to machine precision.

In the case of the ellipse the average relative error in curvature is of 2.77% for 70 particles, and of 0.7% for 140 particles. This error is computed in relation to the analytical expression for the curvature of an ellipse. This provides a sufficiently accurate approximation of the curvature for the purposes of this work, since for most reasonable resolutions, the error is below the 1% threshold we set as acceptable for the density.

## 3.6 Handling Sharp Corners

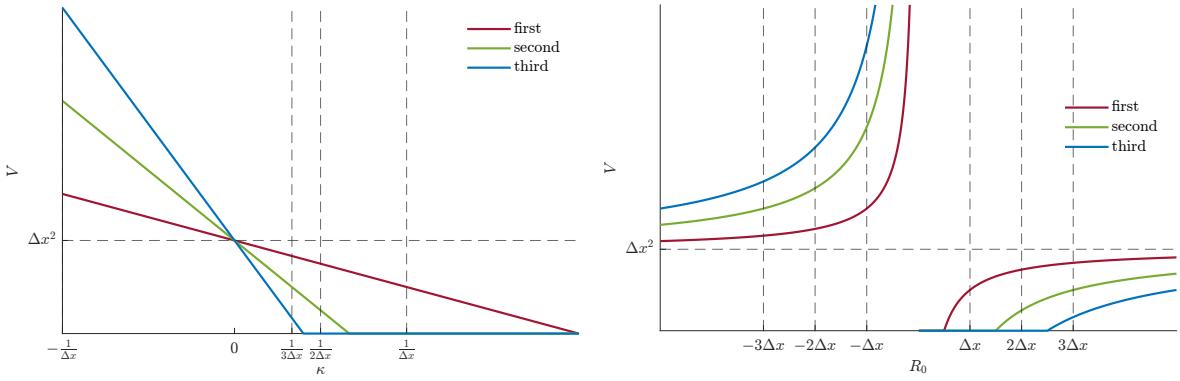
The modelling of sharp corners is problematic since from a geometric point of view the curvature can not be well defined at these points. Since the normal vectors are discontinuous at the sharp corner, the curvature remains undefined. One can consider it to be infinite, since for increasingly sharp corners, the radius of the osculating circle tends to zero, and thus the curvature tends to infinity. This is of course problematic for the present SPH wall model, since the curvature is used to compute the volume of the virtual particles, which intervenes in the calculation of the forces and densities.

Using our methodology to compute the curvature, sharp corners are also problematic. For smooth geometries, with very high curvatures, the curvature will be computed correctly as long as it is below  $\kappa_{\max}$ . Thus if at a given resolution we are not capable of resolving the curvature correctly, an increase of resolution, thus an increase of the maximum curvature, fixes the problem. However when computing

the curvature at a sharp corner, the values are always close to  $\kappa_{\max}$ , no matter the resolution. This is a consequence of the geometrical singularity of the curvature not a flaw of the numerics.

To understand the reasoning behind the maximum curvature allowed, consider volume of the virtual particles in the convex case. The derivation is valid given that  $R_0 \geq 3\Delta x$ . Otherwise, the radius  $R_3$  would be zero for  $R_0 = 3\Delta x$ . This implies an upper bound on the curvature,  $\kappa < \frac{1}{3\Delta x}$ . Beyond this point, the geometrical interpretation is no longer valid, but if one evaluates the volume expression for  $R_0 < 3\Delta x$  (equivalently,  $\kappa > \frac{1}{3\Delta x}$ ), one can see that the volume of the third particle becomes negative at  $R_0 = \frac{5}{2}\Delta x$ , the volume of the second particle becomes negative at  $R_0 = \frac{3}{2}\Delta x$ , and the volume of the first particle becomes negative at  $R_0 = \frac{1}{2}\Delta x$ .

In Fig.3.5 we present a visualization of the behavior of the volume of the virtual particles as a function of the curvature, and the radius of the osculating circle, as given by Eq.3.7. One can see that for negative curvatures (concave), there is no problem since as the curvature goes towards negative infinity, the volume of the particles grows and stays positive. However, the convex case is problematic, since as the curvature grows, the volume of the virtual particles first becomes zero and later negative.



**(a)** Volume of the virtual particles as a function of the curvature. **(b)** Volume of the virtual particles as a function of the radius of the osculating circle.

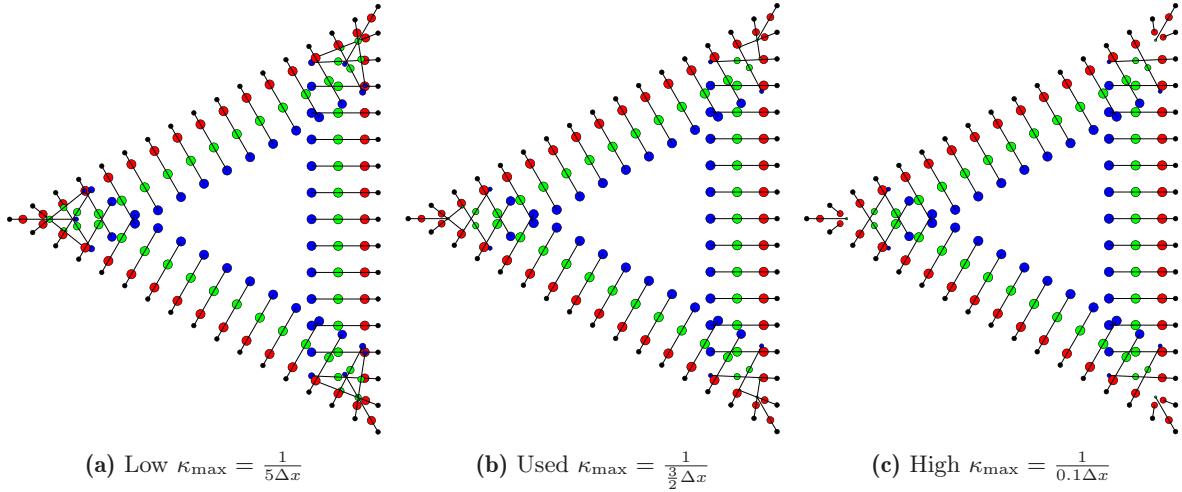
**Figure 3.5:** Volume of the three virtual particles, depending on the curvature/radius of curvature. Following Eq.3.24, setting  $\Delta s = \Delta x$

This motivates us to redefine the volume of the virtual particles as a maximum between zero and the volume computed by Eq.3.7, as

$$V_n = \max(0, \frac{1}{2} (2\Delta x + (1 - 2n) \Delta x^2 \kappa) \Delta s), \quad (3.24)$$

in order to avoid negative volumes. And set an upper limit to the curvature at  $\kappa_{\max} = \frac{1}{3\Delta x}$ , the value at which the volume of the second particle becomes zero. This allows for the volume of the second and third virtual particle to become zero in regions of high curvature where the virtual particles, from neighboring marker particles, overlap and are extremely tightly packed. Also it forces that at any resolution, for both smooth geometries with high curvature, and sharp corners, the first virtual particle has a non-zero positive volume. This is a reasonable choice, since the volume of the first particle is the most important, since being the closest one to the fluid, it has the highest kernel weighting. And in such scenarios of high curvature, the space where the second and third particle would sit is already occupied by the virtual particles of neighboring marker particles, so no region of the support of a nearby fluid particle is left unoccupied.

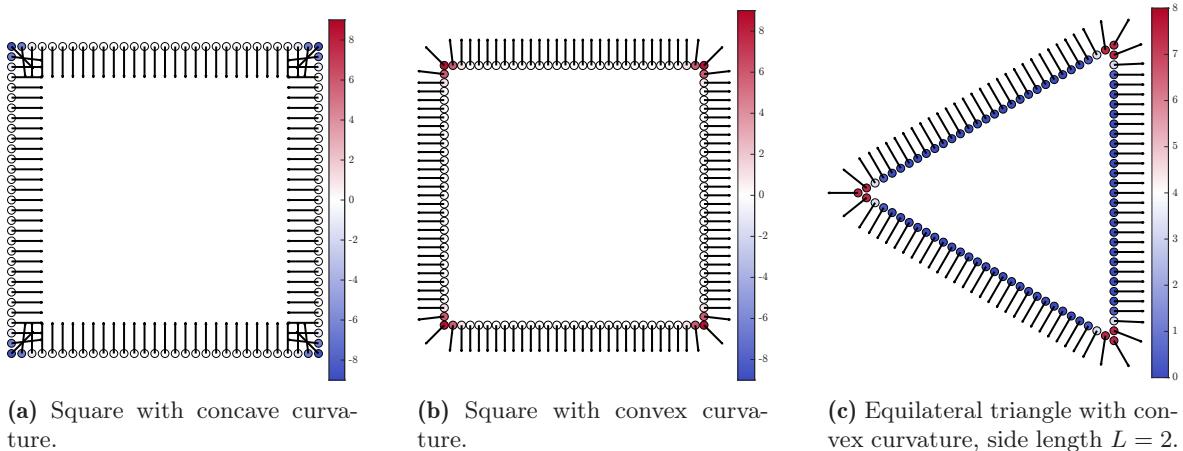
If no threshold were set on the maximum contribution in the summation from Eq.3.23, very high curvatures would be computed at sharp corners, and the volume of their virtual particles would be zero, thus disabling the corner from interacting with the fluid particles. On the other hand, if a not high enough threshold were set, the volume of the second and third virtual particles would always be non-zero at sharp corners, and the virtual particles at the corner would be densely packed, even leaving the solid domain in extreme cases, thus creating a region of high density and pressure, which causes unphysical repulsive forces on the fluid phase. In Fig.3.6 we present a visualization of this effect of the maximum curvature threshold on the volume of the virtual particles at a sharp corner. It can be seen that when a too low value for the maximum curvature is set, the virtual particles at the



**Figure 3.6:** Volume of the virtual particles at a sharp triangular corner, for different values of  $\kappa_{\max}$ . Marker particles are represented in black, first second and third virtual particles are represented in red, green and blue respectively.

corner are very tightly packed. On the other hand when the maximum curvature is set too high, the virtual particles at the corner have near zero volume, and the corner interaction with the fluid phase is too weak. Using the compromise value of  $\kappa_{\max} = \frac{1}{2\Delta x}$ , the virtual particles at the corner have a volume that is neither zero nor too high, and the corner interacts with the fluid phase in a physically reasonable way.

In Fig.3.7 we display the normal vectors and curvatures that our method produces for the case of geometries with sharp corners. It can be seen how the method for computing the normal vectors produces vectors that curve around the corners, making the fluid interact with the corner smoothly. In terms of curvature, sharp concave corners get a negative curvature, which makes the virtual particles in the corner occupy more volume, with respect to a flat wall. In convex corners the curvature is a high positive value close to  $\kappa_{\max}$ , and the virtual particles occupy less volume to avoid the overlap with the virtual particles of neighboring marker particles. In the case of the triangular sharp corner, the curvature is high in the three corner particles, making the virtual volume of their third marker particle zero, and lowering the other two, thus preventing a clustering of virtual particles at the corner.

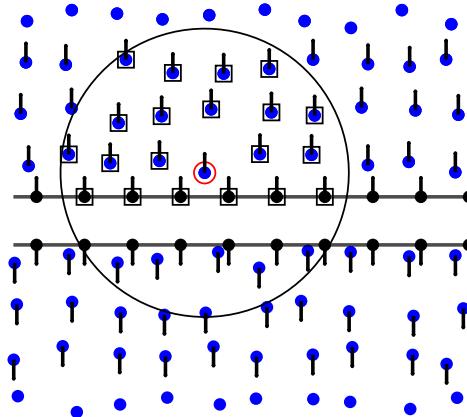


**Figure 3.7:** Examples of numerically computed normal vectors, and curvature for squares with concave and convex corners and an equilateral triangle. The squares have side  $L = 2$ . The normal vectors are represented in black, and the curvature by the color scale. The particle spacing is equal to the marker spacing,  $\Delta s = \Delta x = \frac{1}{15}$  and the maximum curvature is  $\kappa_{\max} = 10$ .

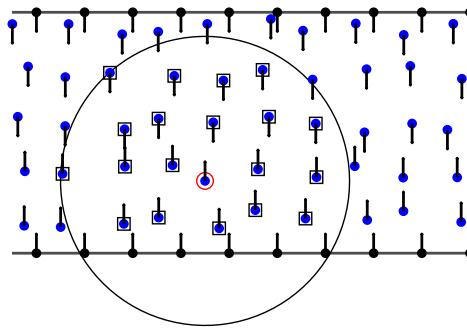
A problematic situation arises to model thin solid profiles, surrounded by both sides by fluid. In this case, where the thickness of the solid is smaller than a kernel radius, fluid particles on one side will interact with marker particles of both sides and fluid particles of the opposite side. This leads to unphysical situations. In other methods, this requires to increase the resolution until the kernel radius is smaller than the thickness of the solid. However, this is not always possible, and might be prohibitively expensive. In the present method, since interaction with the wall is condensed in the marker particles, strategies to identify the side of the wall that the fluid particles are interacting with can be implemented. In the following we present a simple strategy to limit the interaction of the fluid particles with the solid to only one side. This is only a proof of concept to show the possibilities this formulation offers, and more sophisticated strategies could be implemented.

The strategy is inspired by the work of Shadloo et al.[31]. Fluid particles within the interaction distance of a marker particle are assigned the normal vector of the closest marker particle. Other fluid particles get a zero normal vector. This step of assigning normal vectors to the fluid particles is done at every time step after the position update.

Then when fluid particles interact with their neighbors, a check is performed on the normal vectors of every pair of particles to allow the interaction or not. The interaction is allowed only if the angle between normal vectors, computed from the dot product, is smaller than a threshold. In this work we set the maximum angle between normal vectors to  $\theta_{\max} = \pm 100^\circ$ . In Fig.3.8 a diagram is presented to illustrate the concept.



**Figure 3.8:** Diagram illustrating the concept of the thin wall interaction limiter. Fluid particles (blue) close to a marker particles (black) are assigned the normal vector of the closest marker particle. A fluid particle (circled in red) interacts only with particles with aligned normal vectors (marked by squares).



**Figure 3.9:** Diagram illustrating the need to allow interaction of fluid particles with opposed normals in some problematic cases. If the interaction is limited only to particles with aligned normals, the fluid particles in the upper half of the duct would not interact with the ones at the lower half.

Additionally a second check is performed, to solve the problematic case of fluid particles inside parallel walls. In such a case the walls have opposite normals. Such problematic situation is illustrated in Fig.3.9 by a duct with parallel walls.

---

**Algorithm 1** Modification of the interaction algorithm to limit the interaction of fluid particles with thin walls.

---

```

for all fluid particles  $i$  do
    for all neighbors  $j$  of particle  $i$  do
        if  $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j > \cos(\theta_{\max})$  or  $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{r}}_{ji} > \cos(\theta_{\max})$  or  $\hat{\mathbf{n}}_j \cdot \hat{\mathbf{r}}_{ij} > \cos(\theta_{\max})$  then
            Interact( $i, j$ )
        else
            Skip ( $i, j$ ) interaction
        end if
    end for
end for

```

---

If only the previous criterion is applied, fluid particles in the upper half of the duct would not interact with those in the lower half, as they have opposite normals. To avoid this situation, a second check is performed. Interaction is allowed if the unit vector between the two particles  $e_{ij}$  forms an angle smaller than  $\theta_{\max}$  with the normal vector  $\hat{\mathbf{n}}_j$ . The symmetric criterion for  $e_{ji}, \hat{\mathbf{n}}_i$  is also applied, to ensure that the interaction is symmetric. The algorithm is summarized in Algorithm 1.



## Chapter 4

# Comparison With Analytical Solutions

In this section we present the comparison of the solutions obtained with the present SPH method with analytical solutions for some classical fluid flow problems. The purpose of these comparisons is to validate the implementation of the method and examine the convergence of the solution with resolution.

### 4.1 Poiseuille and Couette Flow

The first examples are the classical channel flow problems, Poiseuille and Couette flow. The flow is set in an infinite channel with height  $L_y = 1$ . In the first case, the flow is driven by a body force in the  $x$  direction,  $g = 0.1$ , and in the latter, the flow is driven by the upper wall moving in the  $x$  direction with a constant velocity  $u_w = 0.125$ .

To simulate the channel we set periodic boundary conditions in the  $x$  direction, and a width of  $L_x = 0.4L_y$ . We set  $\rho_0 = 1$ ,  $\nu = 0.1$ ,  $p_B = p_0$ . In both cases we set  $u_{\max} = 0.125$ , since the solution to the Poiseuille flow in the steady state is the well known parabolic profile with  $u_{\max} = \frac{gL_y^2}{8\nu} = 0.125$ , and in the Couette flow, the maximum velocity is that of the upper wall. To compute the density, we use the continuity equation.

The analytical solution for the time evolution of the Poiseuille flow, starting from rest, is given by

$$u_x(y, t) = \frac{g}{2\nu}y(L - y) - \sum_{n=1}^{\infty} \frac{4gL^2}{\nu\pi^3(2n+1)^3} \sin\left(\frac{\pi y}{L}(2n+1)\right) \exp\left(-\frac{(2n+1)^2\pi^2\nu}{L^2}t\right). \quad (4.1)$$

The analytical solution for the Couette flow is given by

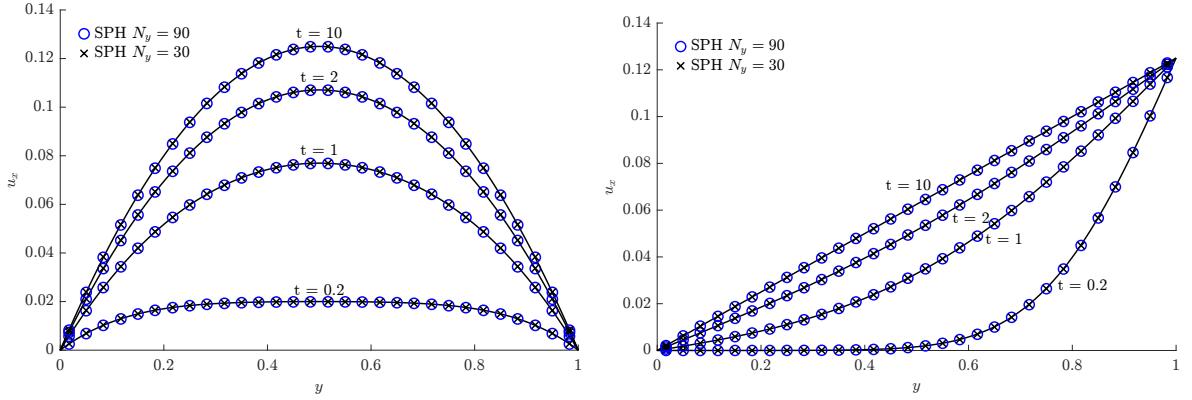
$$u_x(y, t) = \frac{u_w}{L}y + \sum_{n=1}^{\infty} \frac{2U_0}{n\pi} (-1)^n \sin\left(\frac{n\pi}{L}y\right) \exp\left(-\nu\frac{n^2\pi^2}{L^2}t\right). \quad (4.2)$$

Both solutions are written in terms of a steady state solution plus a series of terms that decay exponentially with time. To compute the analytical solution we truncate the series at  $n = 100$ .

In Fig.4.1, we show a comparison of the time evolution between the SPH solution and the analytical solution, for  $t = 0.2, 1, 2, 10$ . The SPH is presented for two resolutions,  $N_y = 30, N_x = 12$  and  $N_y = 90, N_x = 36$ . One can see that the SPH solution reproduces the expected velocity profiles very well, for all time instants and resolutions.

### 4.2 Taylor-Couette Flow

As a second example we choose another flow with analytical solution, albeit with a non-flat geometry, the Taylor-Couette flow. The flow is set between two concentric cylinders, with radius  $R_1 < R_2$ ,



(a) Poiseuille flow analytical solution (solid line) and SPH solution at two resolutions,  $N_y = 30$  and  $N_y = 90$ . In  $N_y = 90$  solution only every third particle is shown for clarity.

(b) Couette flow analytical solution (solid line) and SPH solution at two resolutions,  $N_y = 30$  and  $N_y = 90$ . In  $N_y = 90$  solution only every third particle is shown for clarity.

**Figure 4.1:** Poiseuille and Couette flow comparison between analytical solution and SPH solution. Time evolution from rest to steady state at  $t = 0.2, 1, 2, 10$ .

rotating with angular speeds  $\Omega(R_1) = \Omega_1$  and  $\Omega(R_2) = \Omega_2$ . The steady state analytical solution for the velocity field is given by

$$u_\theta = Ar + \frac{B}{r} \quad \text{with} \quad A = \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2} \quad B = \frac{R_2^2 R_1^2}{R_2^2 - R_1^2} (\Omega_1 - \Omega_2) \quad (4.3)$$

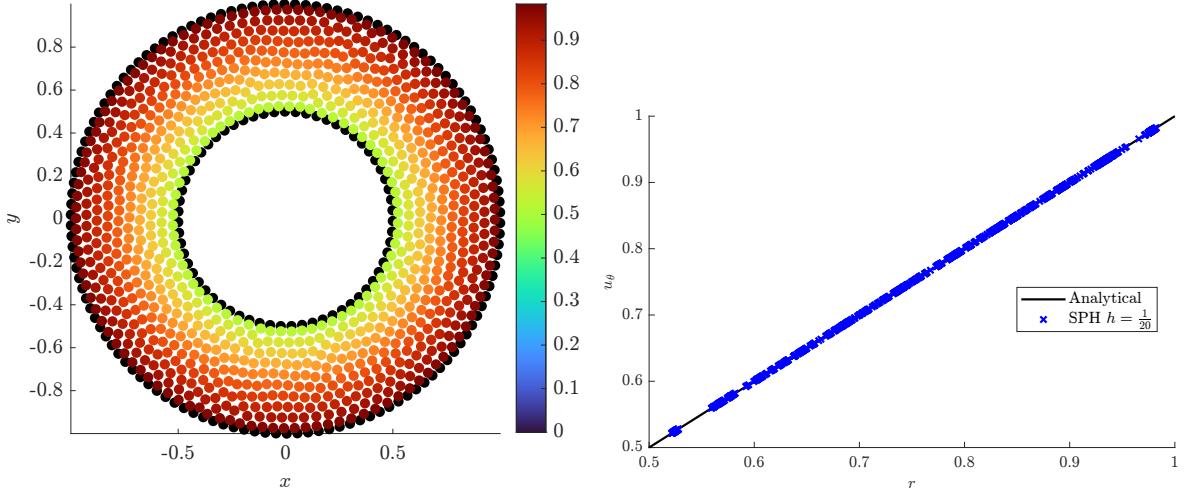
$$u_r = 0.$$

We simulate this flow with  $R_1 = 0.5$ ,  $R_2 = 1$ ,  $\Omega_1 = 1$ ,  $\Omega_2 = 1$ ,  $\rho_0 = 1$ ,  $\nu = 0.1$ ,  $p_B = p_0$ . We fill the inter-cylinder region with fluid particles initially arranged in a cartesian grid with  $\Delta x = h = \frac{1}{h}$ . In Fig.4.2, we show the particle distribution at steady state. And the comparison between the analytical solution and the SPH solution of the  $\theta$  component of the velocity field. The SPH solution reproduces the expected velocity profile accurately. In the velocity profile comparison, we observe that the SPH solution, only at the near wall regions, the solution points cluster at one  $r$  value. Due to particles being arranged in circles around the inner and outer cylinders. For other  $r$  values the points are spread out in the  $u_\theta(r)$  plane, due to particle disorder. In contrast to the channel flows, in which the solution in the  $u_x(y)$  plane forms point clusters of particles at fixed  $y$  positions, since particles remain highly ordered in the  $y$  direction, for the whole simulation.

### 4.3 Convergence

To further study the behavior of these solutions with resolution, we compute the  $L_1$  error norm for the Poiseuille and Taylor-Couette flows as a function of the smoothing length  $h$ . For uniformly spaced particles, the error of the SPH solution decreases with  $h$ , for constant  $\frac{\Delta x}{h}$ , until a limiting discretization error is reached, as shown in Quinlan et al. [29]. For non-uniformly spaced particles, the trend is similar, but the discretization error can even increase for decreasing  $h$  due to the non-uniformity of the particle distribution.

To limit the influence of the discretization error, we limit the analysis to the near wall region where the particle distribution is more ordered. Thus averaging over a smaller number of particles, so the error is not dominated by the non-uniform particle distribution. We define this region for the Poiseuille flow as  $y \in [3h, 1 - 3h]$  and for the Taylor-Couette flow as  $r \in [0.5 + 3h, 1 - 3h]$ . The size is chosen to include one kernel radius per side of the wall. The error norm is then defined as



(a) Taylor-Couette particle distribution at steady state for  $h = \frac{1}{20}$ . Black particles represent marker particles. Fluid particles are colored by their velocity magnitude.

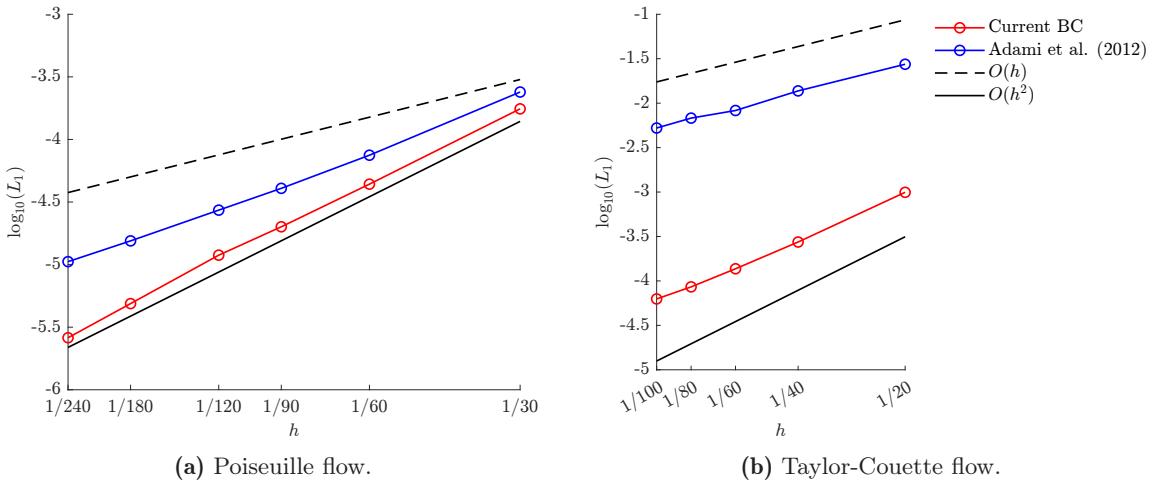
(b) Taylor-Couette comparison between analytical solution (solid line) and SPH solution for  $h = \frac{1}{20}$ .

**Figure 4.2:** Taylor-Couette

$$L_1 = \frac{1}{N} \sum_{\substack{i=1 \\ y \in [3h, 1-3h]}}^N |u_x^{\text{SPH}}(y_i) - u_x^{\text{analytical}}(y_i)| + |u_y^{\text{SPH}}(y_i) - u_y^{\text{analytical}}(y_i)|, \quad (4.4)$$

$$L_1 = \frac{1}{N} \sum_{\substack{i=1 \\ r \in [0.5+3h, 1-3h]}}^N |u_\theta^{\text{SPH}}(r_i) - u_\theta^{\text{analytical}}(r_i)| + |u_r^{\text{SPH}}(r_i) - u_r^{\text{analytical}}(r_i)|. \quad (4.5)$$

For the Poiseuille and Taylor-Couette flows respectively. To compute the analytical solution, we evaluate at the particle positions the steady state analytical solutions given above.



**Figure 4.3:**  $L_1$  error norm for analytical flows in the near wall region. As a function of the smoothing length  $h$ . Lines with first and second order convergence rates are shown for reference.

In Fig.4.3, the error norm of the velocity field is plotted against  $h$ . We observe that the error decreases with  $h$  for both flows. For the Poiseuille flow, we observe a second order convergence rate, while for

the Taylor-Couette flow, the convergence rate is second order for the three coarser resolutions, and roughly 1.7 for the finer ones. This occurs because the Taylor-Couette flow has a non-uniform particle distribution, unlike the highly ordered distribution in the Poiseuille solution. This non-uniformity increases the discretization error for the finer resolutions. Analyzing the entire domain, rather than focusing on the near-wall region, still shows second-order convergence for the coarser resolutions, but the error converges more poorly at finer resolutions.

We also provide a comparison with the boundary condition implementation of Adami et al. [1]. This implementation uses a cartesian placement of particles in the solid region. Solid particles in their method use Shepard interpolation to extrapolate properties from the fluid particles. Then properties are adequately mirrored to enforce the boundary conditions. This implementation differs from ours in two key aspects: the geometric placement of particles and the extrapolation/mirroring procedure. With our extrapolation being done in fluid-solid particle pairs, and the one of Adami et al. being averaged for all nearby fluid particles of a solid particle.

In the Poiseuille case, the flat geometry of the walls makes the two placements equivalent, and the only difference becomes the mirroring procedure. We can see how both methods show near second order convergence, with the current implementation having a slight advantage in the error magnitude. On the other hand for the Taylor-Couette flow, due to the curved walls, there also exists a difference in the particle placement. The cartesian placement of particles does not exactly represent the curved geometry of the walls, the solid will be discretized in a stair-case fashion, with the steps being smaller for finer resolutions. In our case, the marker particles are always placed in a circle, independently of the resolution, and higher resolution only reduces the marker particle spacing along the circle. This difference in the discretization of the solid walls can be seen in the error convergence rate, which unlike in the flat geometry, shows different behavior for the two implementations. The current implementation achieves an average convergence rate of 1.727, primarily due to error plateauing at finer resolutions, whereas coarser resolutions exhibit second-order convergence. In contrast, the Adami et al. implementation, which shows a convergence rate of 1.0289. These findings align with the results of Negi et al. [27] in which the boundary condition implementation of Adami et al. [1] is shown to have second order convergence for flat geometries but first order or worse for curved geometries.

# Chapter 5

## Results

In this section we present the results obtained when applying the present method to more complex fluid flow problems. The studied problems include curved boundaries, sharp corners, moving and accelerating walls and three dimensional problems. Comparison with other reference grid based solutions, and other SPH solutions indicate good agreement and accuracy of the method.

### 5.1 Periodic Array of Cylinders

In this first example, we simulate the flow through a wall bounded periodic array of cylinders. This scenario is represented by a cylinder placed at the centre of a channel, with periodic boundary conditions imposed in the  $x$  direction. The cylinder has radius  $R = 0.02m$  and the channel has height  $L_y = 4R = 0.08m$ , we simulate a domain of length  $L_x = 6R = 0.12m$ . Due to periodicity, the distance between consecutive cylinder centers is  $6R$ . The fluid phase has density  $\rho = 1000kg/m^3$ , kinematic viscosity  $\nu = 10^{-4}m^2/s$ , and  $p_B = p_0$ .

The problem of the flow through a periodic array of cylinders for WCSPH is studied in detail by Ellero and Adams in [8], and also used as validation in the work of Adami et al. [2] where the transport velocity formulation is introduced. We follow the approach of the latter work in which a constant body force  $g = 2.5 \cdot 10^{-4}$  is used to get an average flow velocity of approximately  $\langle u_x \rangle = 1.2 \cdot 10^{-4}m/s$ , as in the work of Ellero and Adams. The a posteriori measured maximum velocity magnitude is  $u_{\max} = 3.6 \cdot 10^{-4}m/s$ , and therefore the sound speed in this case is set through the body force criterion as  $c = 10\sqrt{|g|R} = 2.236 \cdot 10^{-2}m/s$ .

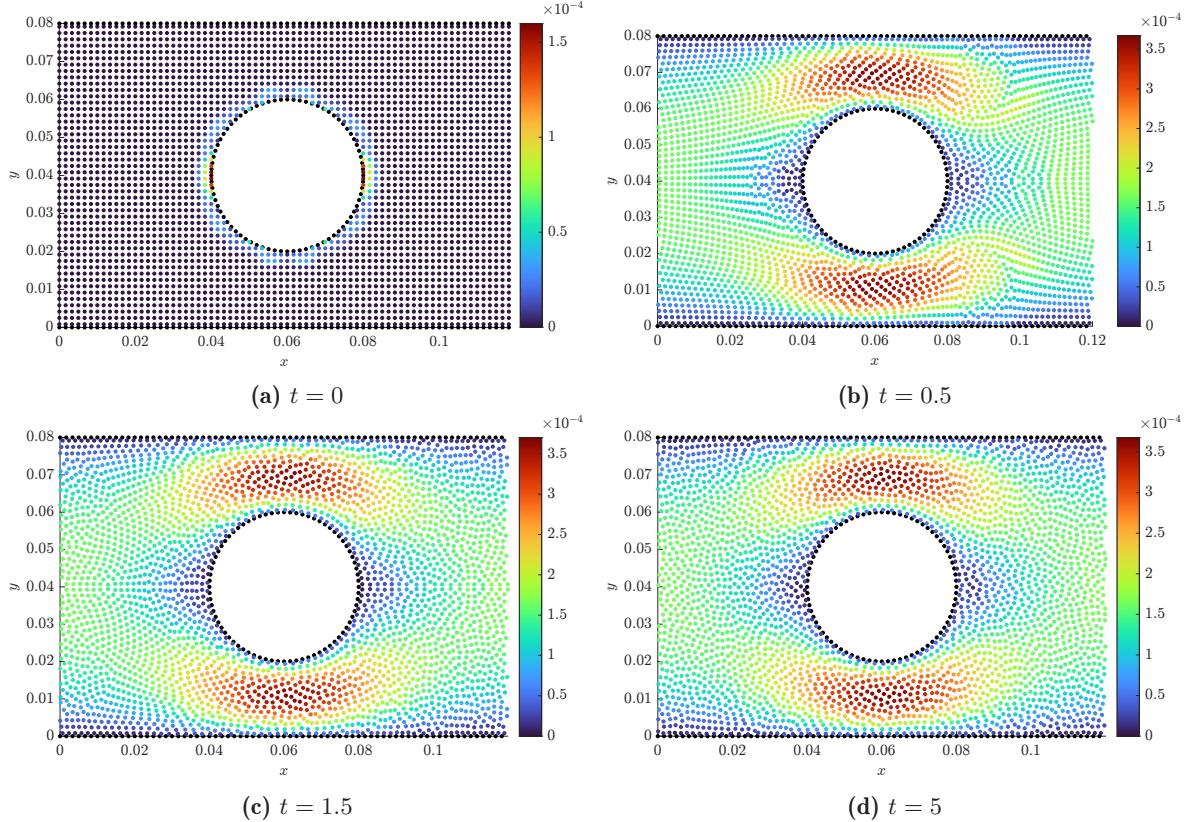
To fill the domain with fluid particles we place the particles in a cartesian lattice, with equal spacing in  $x$  and  $y$  direction, and skip those that are inside the cylinder. The resolutions we use are  $N_y = 48, 96, 192$  which imply  $N_x = 72, 144, 288$  respectively. To place the marker particles we need to determine how many and what arc length element belongs to each one. To use a resolution similar to the fluid phase, we determine the number of particles as the ceil function of the ratio between the perimeter of the cylinder and the fluid particle spacing

$$N = \lceil \frac{2\pi R}{\Delta x} \rceil. \quad (5.1)$$

This number of particles implies an arc length element per particle,  $\Delta s$ , close to but not equal to  $\Delta x$ , since in general  $\Delta x$  does not divide the perimeter of the cylinder. The arc length element is then given by

$$\Delta s = \frac{2\pi R}{N}. \quad (5.2)$$

In Fig.5.1 we show the particle distribution and velocity magnitude at different time instants. It can be seen that due to the cartesian placement of the fluid particles, at the initial instant, some particles are placed too close to the cylinder. This creates strong repulsive forces, that equilibrate the particle



**Figure 5.1:** Particle distribution at different time instants for the periodic array of cylinders. The color scale represents the magnitude of the velocity, marker particles are colored black.

configuration in a few time steps. After approximately  $t = 1.5$  the particle distribution becomes homogeneous and the steady state solution is reached.

The velocity magnitude results shown in Fig.5.1 agree with the SPH results of Ellero and Adams, and Adami et al. To further validate the simulation, we compare the steady-state drag coefficient of the cylinder, with the FEM result of Liu et al.[18] which has a value of  $C_d = 106.76$ . The drag coefficient is computed as

$$C_d = \frac{F_x}{\eta \langle u_x \rangle}. \quad (5.3)$$

Where  $F_x$  is the total force in  $x$  direction acting on the cylinder. In SPH methods, to measure this forces, it is enough to sum all the forces acting on the solid particles. In the present method, to measure this force we accumulate all the forces in the  $x$  direction that the virtual particles exert on the fluid particles, with a change of sign. In Fig.5.2 we show the measured drag coefficient as a function of time for simulations with increasing resolution. It can be seen how after the initial transient, the measured drag coefficient converges to the FEM result. The reached steady state value is closer to the reference solution for increased resolution. The observed initial overshoot is due to the strong forces that rearrange the particle distribution, until a homogeneous particle distribution is reached in steady state.

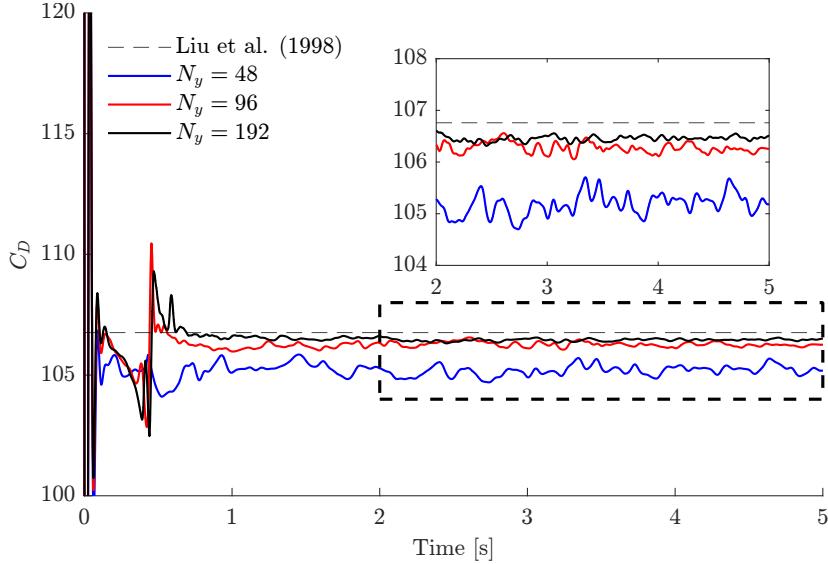


Figure 5.2: Drag coefficient

Resolution	$C_d$
$N_y = 48$	105.18
$N_y = 96$	106.28
$N_y = 192$	106.45
Reference	106.76

Table 5.1: Steady state average drag coefficient for the periodic array of cylinders.

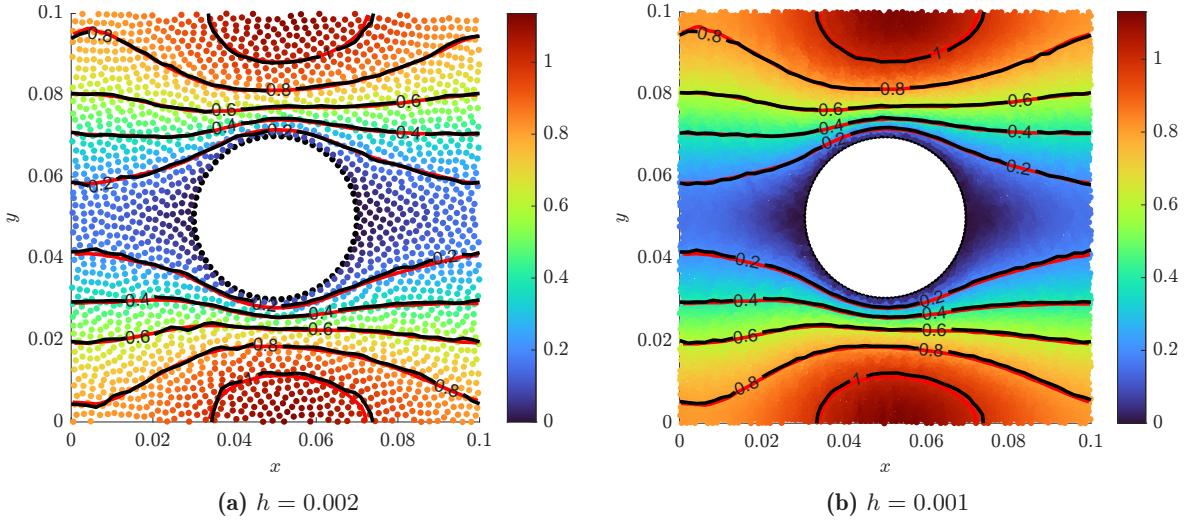
## 5.2 Periodic Lattice of Cylinders

In this scenario we simulate the flow through a periodic lattice of cylinders, following the example from Morris et al. [26]. The simulation domain is a square box of side  $L = 0.1m$  with a cylinder of radius  $R = 0.2m$  placed at the centre. Periodic boundary conditions are set in the  $x$  and  $y$  direction. The fluid phase has density  $\rho = 1000kg/m^3$ , kinematic viscosity  $\nu = 10^{-6}m^2/s$ , and  $p_B = p_0$ . The flow is driven by a body force  $g = 1.5 \cdot 10^{-7}m/s^2$  in the  $x$  direction, and the maximum velocity is  $u_{\max} = 1.2 \cdot 10^{-4}m/s$ , which sets a sound speed  $c = 10u_{\max} = 1.2 \cdot 10^{-3}m/s$ . Due to periodicity this problem models the flow through an infinite lattice of cylinders.

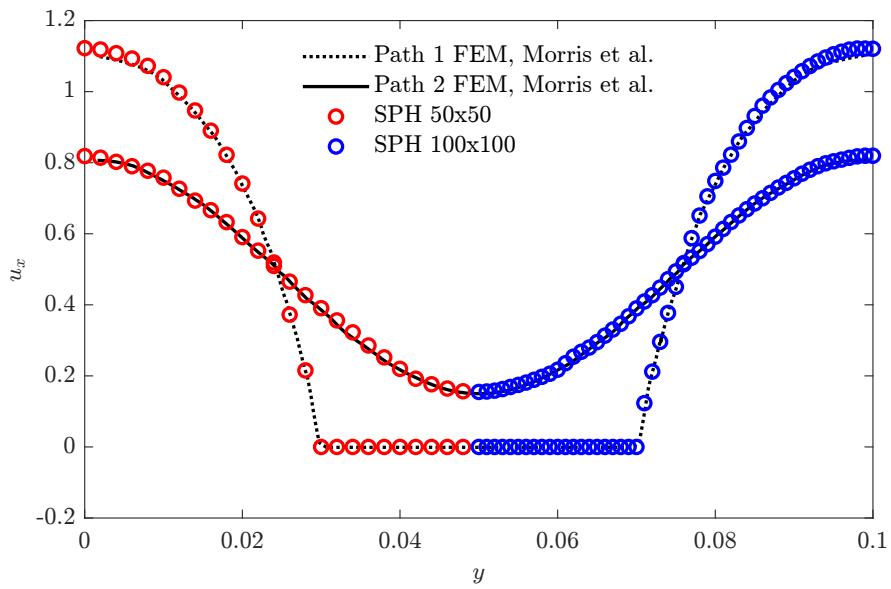
Once again, similarly to the previous example we place fluid particles in a cartesian grid, with equal spacing in both coordinate directions, and omit particles in the space occupied by the cylinder. The marker particles are placed along the cylinder perimeter, with the same placement method as in the previous example. The resolutions we study are  $h = 0.002m$  and  $h = 0.001m$  which correspond to a grid of  $50 \times 50$  and  $100 \times 100$  respectively. This initialization creates 2191 fluid particles and 63 marker particles for the lower resolution and 8751 and 126 for the higher.

In Fig.5.3 we present a comparison of the results of the present method with the FEM reference results presented by Morris et al. [26]. The steady state particle distribution is shown, colored by the velocity magnitude, and superimposed are the contour lines of the velocity magnitude of both our method and the FEM reference. The results show good agreement between the two methods, with the velocity magnitude contour lines matching closely for both resolutions.

Additionally we also present the measured the streamwise velocity component  $u_x$  along two vertical probe lines placed at half the domain length,  $x = 0.5L$  and at the domain end,  $x = L$ . These lines correspond to path 1 and 2 in Morris work. The results are shown in Fig.5.4. The measured streamwise velocity agrees with the FEM reference solution, for path 1 it shows higher velocity values away from the cylinder and lower values near the cylinder. And for path 2 it shows a higher velocity at the channel center line, and lower values towards the vertical ends of the domain.



**Figure 5.3:** Steady state particle distribution, colored by velocity magnitude, for the periodic lattice of cylinders. Superimposed are the contour lines of the velocity magnitude. Red lines represent FEM solution from Morris et al. [26]. Black lines represent the SPH solution. Velocities are scaled by a factor  $u_{\text{ref}} = 10^{-4}$

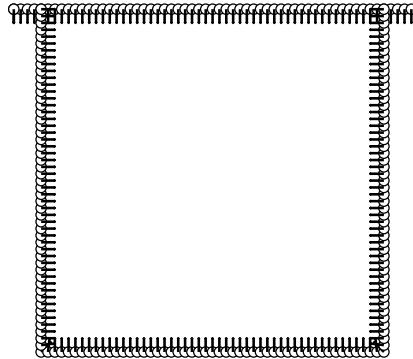


**Figure 5.4:** Measured streamwise velocity,  $u_x$ , at probes placed at vertical lines  $x = 0.5L$  (path 1), and  $x = L$  (path 2). Only one resolution is shown per half path due to symmetry. Solid lines represent FEM solution.

### 5.3 Lid-Driven Cavity

In this section we study a widely used challenging benchmark for SPH solvers the lid-driven cavity problem. The problem is set up as a square box of side  $L = 1$ , with the upper wall moving at constant velocity  $U = 1$  that drives the flow. The fluid phase has a reference density  $\rho_0 = 1$ , sound speed  $c = 10U$  and  $p_B = p_0$ . The Reynolds number is defined as  $Re = UL/\nu$ , we study the cases  $Re = 100, 1000, 10000$  by varying the kinematic viscosity  $\nu$ .

To represent the moving top wall, marker particles are placed along the lid, extended by a distance  $3h$  on its left and right sides, and periodic boundary conditions are set in the  $x$  direction at the lid ends. This extra spacing allows for the the wall to be continuously moving to the right, and avoids the periodic copies of the wall to interact with the fluid. There is an overlap of marker particles at the upper corners of the cavity, between the lateral and upper walls, however their virtual particles do not overlap since the lateral walls have horizontal normals, while the upper wall has a vertical normal.



**Figure 5.5:** Lid-driven cavity problem, marker particle setup, with normal vectors displayed.

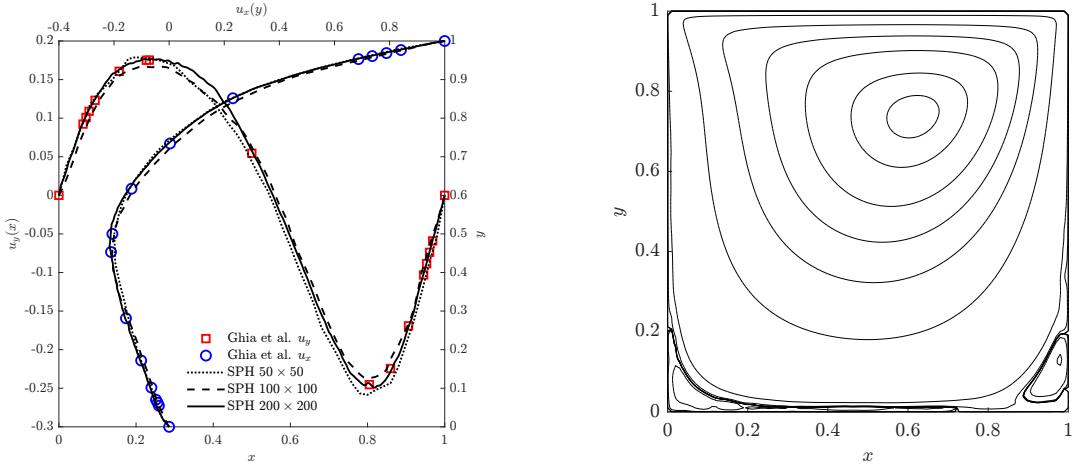
Since no analytical solution exists for this problem we use as reference the results of Ghia et al.[10], who used a multi-grid finite-difference method. We study different resolutions of  $N \times N$  fluid particles, with  $N = 50, 100, 200$ , initially placed in a cartesian lattice. To analyze the steady state of the flow, we monitor the total kinetic energy of the fluid phase, and extract results after the kinetic energy becomes constant.

For  $Re = 100$  the flow is characterized by a single large vortex at the center of the cavity. While for  $Re = 1000$  and  $Re = 10000$  the flow is more complex, with a more intense main vortex and the appearance of secondary vortices in both lower and upper right corners of the cavity. In Fig.5.6 to 5.8 we display the flow streamlines and the measured  $u, v$  velocity profiles at the vertical and horizontal centerlines. For the lowest Reynolds number  $Re = 100$  the measured profiles agree with the reference for all resolutions. While for the higher Reynolds, only the higher resolutions are able to capture the flow features. In the  $Re = 10000$  case, the  $200 \times 200$  resolution is able to capture the secondary vortices, however higher resolution would be needed to exactly match the measured velocity profiles, for example the  $400 \times 400$  resolution used by Adami et al. [2].

The streamlines are displayed only for the  $200 \times 200$  resolution. To compute them, we solve the stream function equation and plot its level curves. To do so we first interpolate the SPH vorticity data to a cartesian grid of the same resolution and interpolate using the cubic kernel. Then we solve the Poisson equation for the stream function  $\psi$

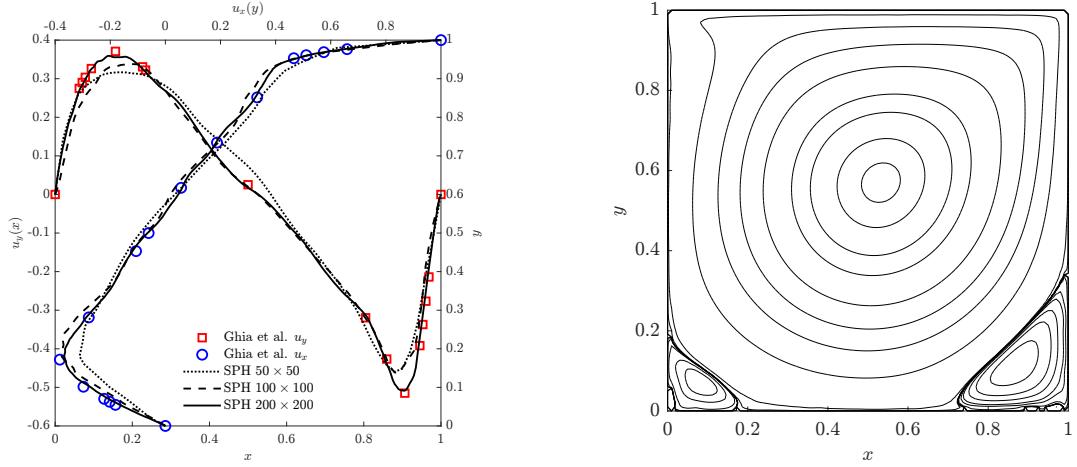
$$\nabla^2 \psi = -\omega, \quad (5.4)$$

discretized as a finite difference equation, and solve it using the Jacobi method. The streamlines are then plotted as the level curves of the stream function.



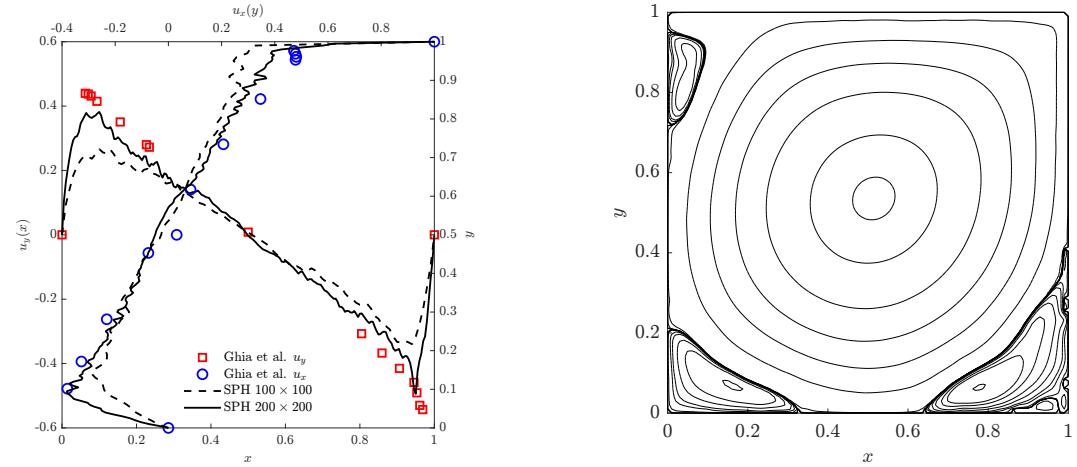
(a) Measured velocity profiles at the vertical and horizontal centerlines. (b) Streamlines for the 200x200 particle simulation.

**Figure 5.6:** Lid driven cavity  $Re = 100$



(a) Measured velocity profiles at the vertical and horizontal centerlines. (b) Streamlines for the 200x200 particle simulation.

**Figure 5.7:** Lid driven cavity  $Re = 1000$

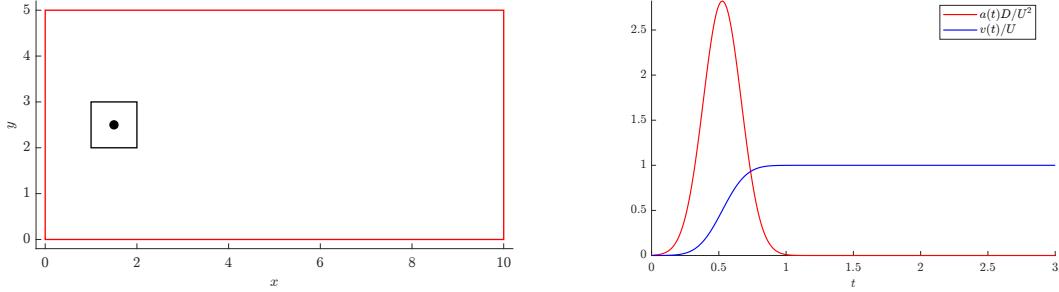


(a) Measured velocity profiles at the vertical and horizontal centerlines. (b) Streamlines for the 200x200 particle simulation.

**Figure 5.8:** Lid driven cavity  $Re = 10000$

## 5.4 Flow Around a Moving Square Inside a Rectangular Box

In the following we study the problem of a moving square inside a rectangular box. The problem is defined in the spheric benchmark case 6 [4]. It is a test case for the validation of separated flows, widely used in the past to validate Large Eddy Simulation (LES) models, and a tough case for SPH solvers. The solver must be capable of generating, diffusing and convecting the intense vorticity around the sharp corners. The benchmark case provides reference results obtained using an incompressible finite difference level set method [5] on a  $600 \times 300$  grid.



**Figure 5.9:** Moving square inside a rectangular box. Problem setup and time law of the motion.

The setup of the problem is detailed in Fig.5.9. The domain is a rectangular box of dimensions  $L_x = 10$ ,  $L_y = 5$ . The origin is placed at the lower left corner of the box, and a square of side  $D = 1$  initially at rest is placed at  $x = 1.5$ ,  $y = 2.5$ . The square is accelerated in the  $x$  direction following the time law shown in the figure.

The acceleration data is provided in the benchmark case for discrete time steps. To be able to evaluate the acceleration at any arbitrary time, we fit a gaussian function to the data, and find excellent agreement. The fitted acceleration function is given by

$$a(t) = Ae^{-\frac{(t-\mu)^2}{2\sigma^2}}. \quad (5.5)$$

The parameters of the gaussian acceleration function are given in Table 5.2. This acceleration smoothly accelerates the square from rest until a maximum velocity  $U = 1$  is reached.

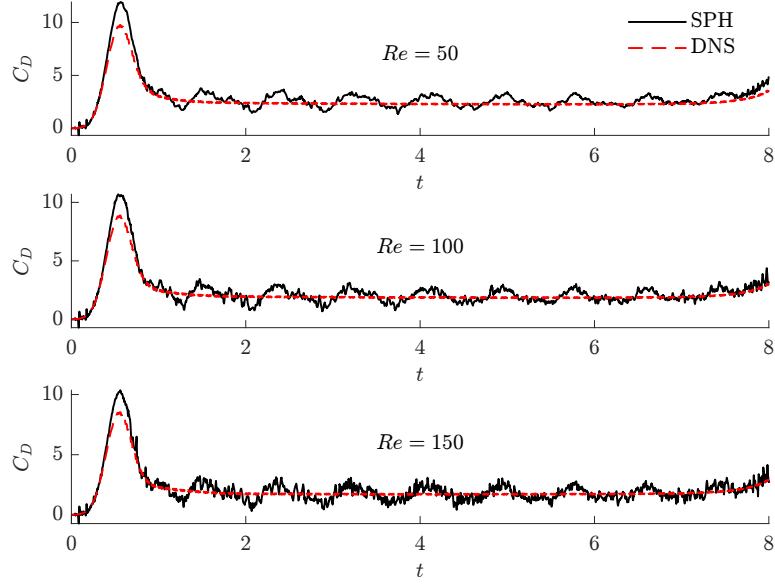
No-slip boundary conditions are set at the walls of the box, and the square. The fluid phase has density  $\rho = 1$ , and  $p_B = 1.5p_0$ . The Reynolds number is defined as  $Re = \frac{UD}{\nu}$ , and we study the cases  $Re = 50, 100, 150$  by varying the kinematic viscosity  $\nu$ . The sound speed is set to  $c = 25$  for all Reynolds numbers. For all cases, the fluid is discretized with  $N_x = 600$  and  $N_y = 300$  particles, initially placed in a cartesian lattice, omitting the space occupied by the square. The resolution is chosen to match the reference grid size.

As a consequence of using the weakly compressible approach, acoustic waves are generated due to the almost impulsive motion of the square, something that would not happen in a truly incompressible fluid. To avoid this side effect a WCSPH method, has to filter the acoustic waves, or make domain boundaries permeable to get rid of them. In this work we do not adopt any of these strategies, and let the acoustic waves propagate freely to not increase the complexity of the method, and focus on the correct enforcement of the solid boundary conditions.

In Fig.5.10 we show the drag coefficient as a function of time for the three Reynolds numbers, compared

Parameter	Value
$A$	$2\sqrt{2}$
$\sigma$	$\frac{1}{4\sqrt{\pi}}$
$\mu$	0.525652

**Table 5.2:** Parameters of the gaussian acceleration function for the moving square.



**Figure 5.10:** Drag coefficient for the moving square inside a rectangular box, for the three considered Reynolds numbers.

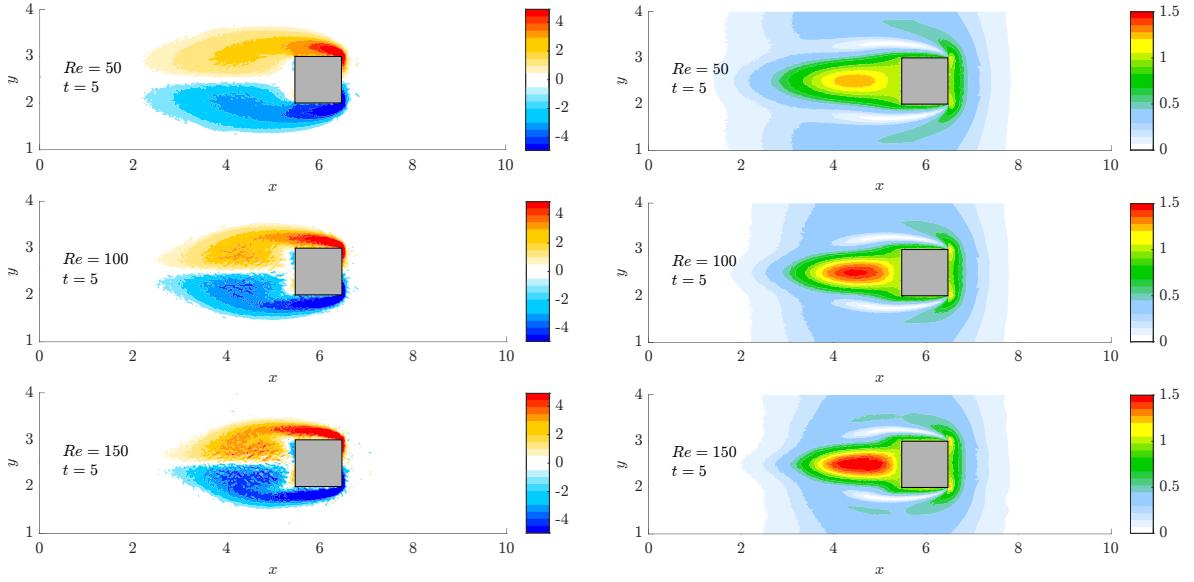
with the reference solution. In this case, the drag coefficient is computed as

$$C_d = \frac{|F_x|}{\frac{1}{2}\rho U^2 D}, \quad (5.6)$$

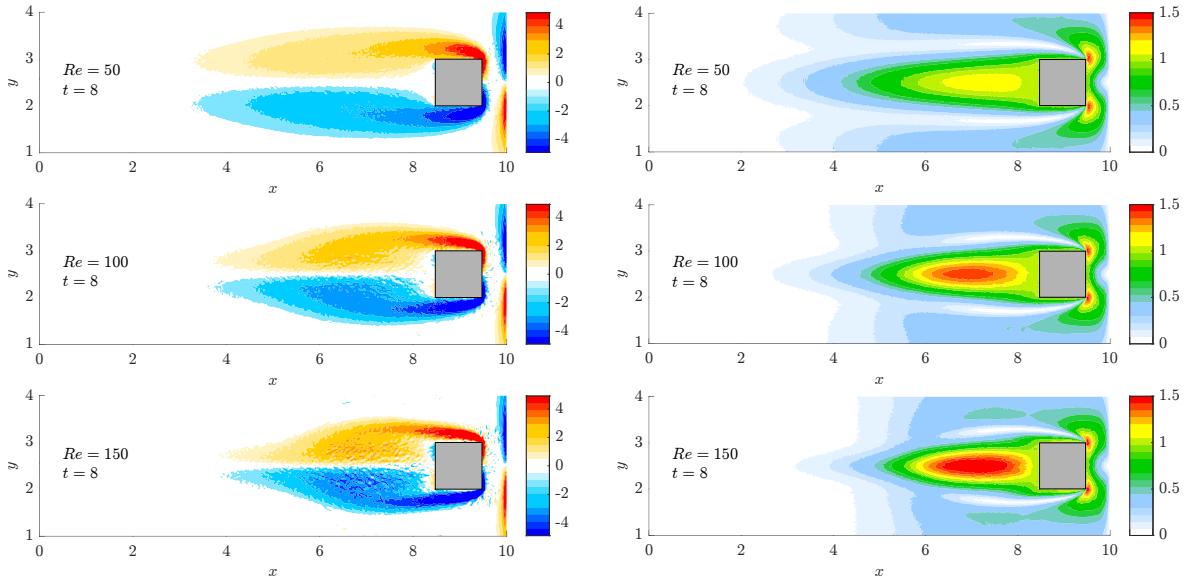
to use the same definition as the reference results. The total horizontal force acting on the square, is computed as in the array of cylinders case, by accumulating the forces in the  $x$  direction that the virtual particles exert on the fluid particles, with a change of sign.

The drag coefficient obtained with the SPH solution agrees with the reference results for all Reynolds numbers, except for the oscillations due to the acoustic waves. There is also a difference in the predicted maximum drag, which is higher in the SPH solution. This discrepancy is also explained by compressibility effects. As per the work of Parmar et al. [28], the Mach number of  $Ma = 0.1$  we use in the WCSph method causes an overshoot of about 10% in the maximum drag coefficient. In the work of Marrone et al. [22] this overshoot is also acknowledged for the same problem simulated with WCSph method, and shown to be reduced with increasing Mach number.

In Fig. 5.11 and 5.12 we show the vorticity and velocity magnitude contours at  $t = 5$  and  $t = 8$  for the three Reynolds numbers. To plot the contours, the data is first interpolated to a cartesian grid of the same resolution, using the cubic kernel as interpolation function. The presented velocity magnitude results agree well with the reference results of the benchmark case, showing the highest velocities at the wake of the square, increasing in magnitude with the Reynolds number. The vorticity contours also agree with the reference results, showing intense vorticity being generated at the front corners of the square, and being convected to the back of the square. The vorticity contours are more diffused for the lower Reynolds number, and more intense for the higher Reynolds numbers, as expected. However the vorticity contours are affected by noise, and can not represent the exact sharp contour lines of the reference results, due to particle disorder around the square corners, and the smoothing effect of the SPH method.



**Figure 5.11:** Vorticity and velocity magnitude contours at  $t = 5$  for the moving square inside a rectangular box, for the three considered Reynolds numbers.



**Figure 5.12:** Vorticity and velocity magnitude contours at  $t = 8$  for the moving square inside a rectangular box, for the three considered Reynolds numbers.

In this section we study two problems of flow around a cylinder of different shapes, elliptical and triangular. These geometries provide a good test case for solid surfaces with non-constant curvature and sharp corners, respectively. The problem is also studied by Antuono et al. [3] using the  $\delta$ -SPH method, and a finite volume method.

As elliptical cylinder we use an ellipse with major axis  $D = 1$  and minor axis 0.4, with the major axis tilted by  $20^\circ$  with respect to the  $x$  axis. The triangular cylinder is an equilateral triangle with side  $D = 1$ , with one vertex oriented in the negative  $x$  direction. The cylinders are placed at the origin, inside a channel that extends from  $x = -6D$  to  $x = 12D$ , and from  $y = -4D$  to  $y = 4D$ . We impose free-slip conditions on the top and bottom walls of the channel, inflow and outflow conditions at the left and right ends of the channel, and no-slip boundary conditions on the cylinders. The fluid phase has density  $\rho = 1$ , and  $p_B = 1.5p_0$ . The Reynolds number is defined as  $Re = \frac{UD}{\nu}$ , and we study the cases  $Re = 100, 1000$  by varying the kinematic viscosity  $\nu$ .

The fluid is initially at rest, and the velocity of the inflow is smoothly increased to  $U = 1$  in a time interval of  $\Delta t = 4$ . The time law for the inflow velocity is given by

$$u(t) = U \begin{cases} (\frac{t}{4})^2(3 - 2\frac{t}{4}) & \text{if } t \leq 4 \\ 1 & \text{if } t > 4 \end{cases} \quad (5.7)$$

To implement inflow outflow conditions, we actually implement periodic boundary conditions in the  $x$  direction, and impose the velocity  $(u(t), 0)$  for the fluid particles at a small buffer region of thickness  $5h$  at the left end of the channel. Although this approach does not represent true inflow, outflow conditions for particle simulations, it allows for a simple implementation, and allows obtaining the expected features and qualitative results of the problem. The drawback is that the inflow has some influence on the right end of the domain, and this causes strong particle disorder in the acceleration phase, introducing some noise in the results.

We measure the drag and lift coefficients, and the Strouhal number of the cylinders, defined as

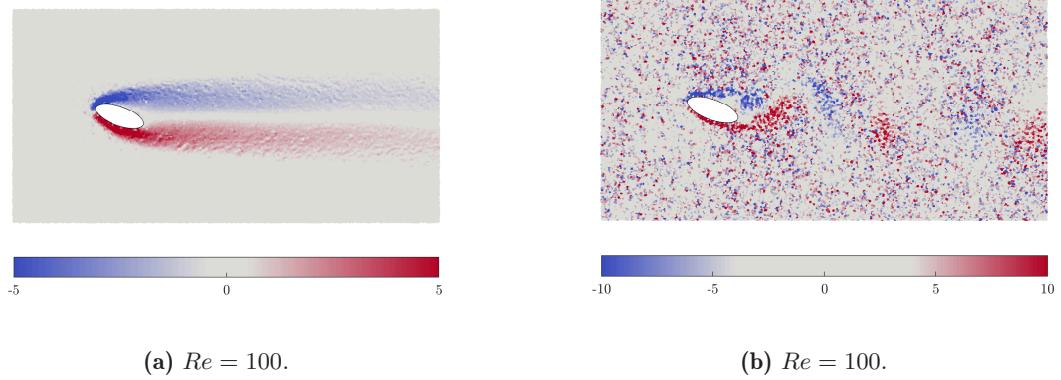
$$\begin{aligned} C_d &= \frac{F_x}{\frac{1}{2}\rho U^2 D}, \\ C_l &= \frac{F_y}{\frac{1}{2}\rho U^2 D}, \\ St &= \frac{fD}{U}. \end{aligned} \quad (5.8)$$

Where  $f$  is the vortex shedding frequency, computed as the inverse of the period between two consecutive peaks of the lift coefficient. The drag and lift forces are computed as in the previous cases, by accumulating the forces in the  $x$  and  $y$  direction that the virtual particles exert on the fluid particles, with a change of sign.

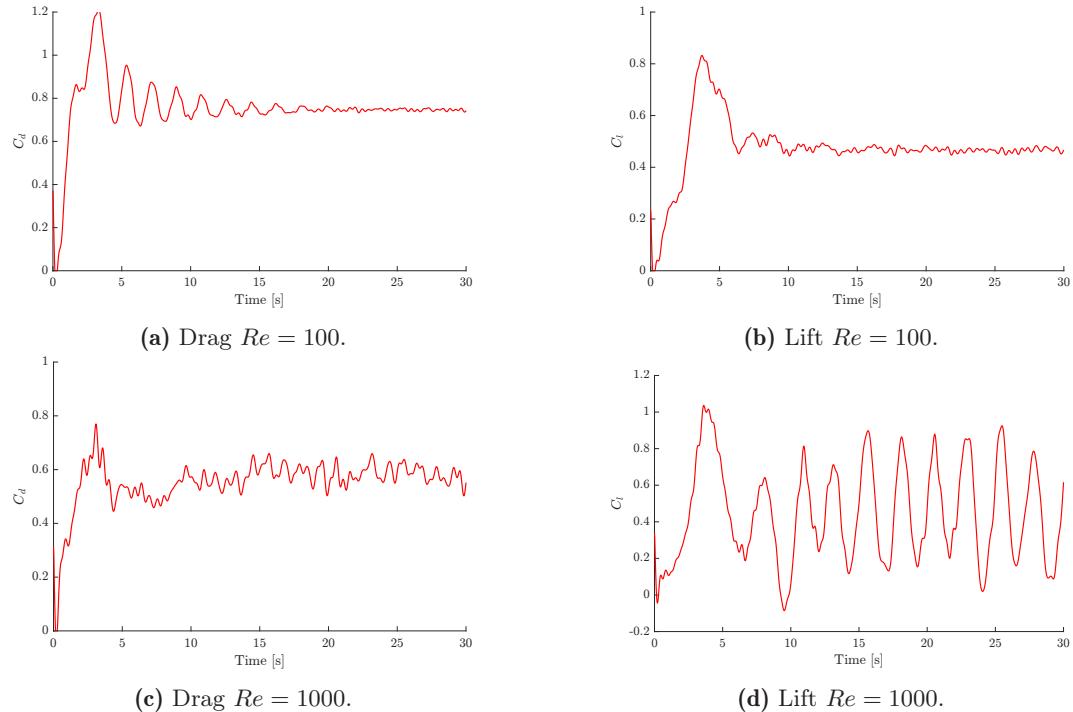
In Fig.5.13 we display the vorticity field around the elliptical cylinder at  $Re = 100$ , and  $Re = 1000$ . At  $Re = 100$  there is no vortex shedding, and the vorticity field is strong around the cylinder and it diffuses downstream. At  $Re = 1000$  the flow is characterized by vortex shedding, and the vorticity field shows the formation of vortices at the back of the cylinder, and their convective motion downstream. With an alternating pattern between the upper and lower vortices.

For  $Re = 1000$  the vorticity field shows the formation of vortices at the back of the cylinder, and their convective motion downstream. The vorticity field is affected by the inflow conditions, and the particle disorder in the acceleration phase, which causes noise in the vorticity field, which becomes more apparent for  $Re = 1000$  due to the lower viscosity. Other works, like the one of Antuono et al. [3] show a cleaner vorticity field, due to the use of a more sophisticated inflow outflow condition and fluid SPH solver.

However the values obtained for the drag and lift coefficients, which are integrated quantities, are in good agreement with the reference results. These results are presented in Table 5.3 and 5.4. Additionally the Strouhal number at  $Re = 1000$  is also in good agreement with the reference results. In Fig.5.14 we also show the drag and lift coefficients time series.



**Figure 5.13:** Vorticity field for the elliptical cylinder at  $Re = 100$  and  $Re = 1000$ .



**Figure 5.14:** Drag and lift coefficients as function of time for the elliptical cylinder at the  $N = 50$  resolution.

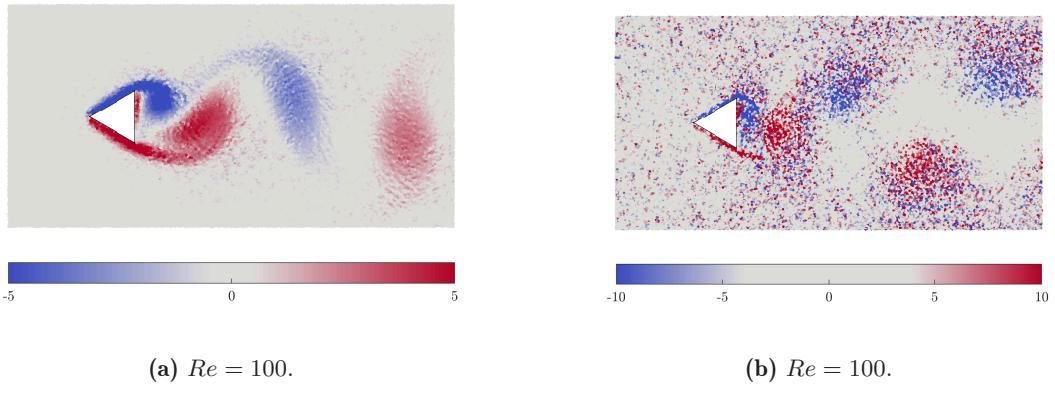
	$C_d$	$C_l$
$N = 25$	0.760	0.469
$N = 50$	0.748	0.467
FVM Antuono et al.	0.730	0.464

**Table 5.3:** Drag and lift coefficients, steady-state averaged values for the elliptical cylinder at  $Re = 100$ .

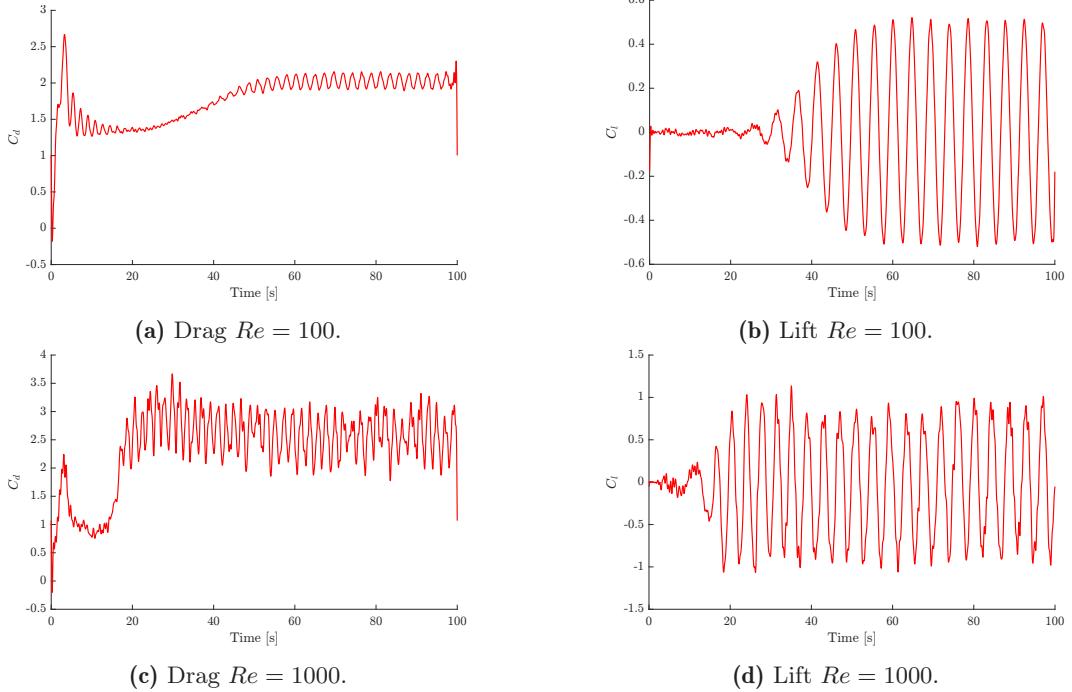
	$C_d$	$C_l$	St
$N = 25$	$0.608 \pm 0.125$	$0.491 \pm 0.341$	0.346
$N = 50$	$0.587 \pm 0.060$	$0.462 \pm 0.376$	0.402
FVM Antuono et al.	$0.519 \pm 0.041$	$0.482 \pm 0.425$	0.446

**Table 5.4:** Drag and lift coefficients, average and amplitude, for the elliptical cylinder at  $Re = 1000$ .

With respect to the triangular cylinder, the vorticity field is shown in Fig.5.15 for  $Re = 100$  and  $Re = 1000$ . In both cases we observe vortex shedding, with the formation of vortices at the back of the cylinder, and their convective motion downstream. The vorticity field is also affected by the inflow conditions, and the particle disorder in the acceleration phase, which causes noise in the vorticity field for  $Re = 1000$ . The drag and lift coefficients are presented in Table 5.5 and 5.6, and the time series of the drag and lift coefficients are shown in Fig.5.16. Good agreement with the reference results is obtained for the drag and lift coefficients, and the Strouhal number at both Reynolds numbers.



**Figure 5.15:** Vorticity field for the triangular cylinder at  $Re = 100$  and  $Re = 1000$ . Snapshots at maximum lift.



**Figure 5.16:** Drag and lift coefficients as function of time for the triangular cylinder at the  $N = 50$  resolution.

	<b>C<sub>d</sub></b>	<b>C<sub>l</sub></b>	<b>St</b>
$N = 25$	$1.985 \pm 0.108$	$\pm 0.480$	0.216
$N = 50$	$2.026 \pm 0.109$	$\pm 0.502$	0.216
FVM Antuono et al.	$1.95 \pm 0.075$	$\pm 0.473$	0.214

**Table 5.5:** Drag and lift coefficients for the triangular cylinder at  $Re = 100$ .

	<b>C<sub>d</sub></b>	<b>C<sub>l</sub></b>	<b>St</b>
$N = 25$	$2.122 \pm 0.340$	$\pm 0.798$	0.264
$N = 50$	$2.570 \pm 0.452$	$\pm 0.877$	0.238
FVM Antuono et al.	$2.33 \pm 0.330$	$\pm 0.650$	0.213

**Table 5.6:** Drag and lift coefficients for the triangular cylinder at  $Re = 1000$ .

## 5.6 Flow Over a Sphere

To show the direct applicability of the boundary condition to three dimensional geometries, we simulate the problem of the flow around a sphere. The method is the same in three dimensions, only the expression for the virtual particles volume is modified to account for the three-dimensional case. Additionally, now the projection of the velocity on the tangential direction is not done just in a single vector, but in a pair of linearly independent vectors, to account for the two tangential directions in three dimensions.

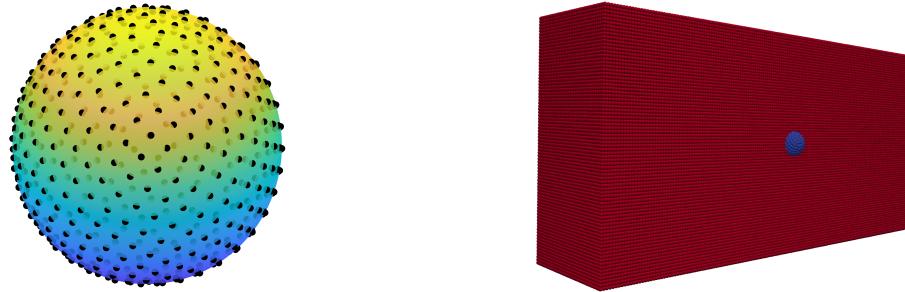
To discretize the sphere, we place particles using the Fibonacci lattice method [12]. This approach generates a quasi-uniform distribution of points on the sphere surface. For a set of  $N$  points, the latitude and longitude angles of the  $i$ -th point are given by

$$\begin{aligned}\lambda &= \arcsin\left(\frac{2i - N - 1}{N}\right), \\ \theta &= \frac{2\pi i}{\phi}.\end{aligned}\tag{5.9}$$

Where  $\phi = \frac{1+\sqrt{5}}{2}$  is the golden ratio. The cartesian coordinates of the particle are then given by

$$\begin{aligned}x &= R \cos(\theta) \cos(\lambda), \\ y &= R \sin(\theta) \cos(\lambda), \\ z &= R \sin(\lambda).\end{aligned}\tag{5.10}$$

In Fig.5.17a we show the points distributed on the sphere surface using this method. The obtained distribution is quasi uniform, and for our purposes it is assumed to be uniform, each particle representing a constant volume element of the sphere.



(a) Points distributed on the sphere surface using the Fibonacci lattice method.  
(b) Fluid (red) and marker (blue) particles initialization. Fluid particles clipped for visualization of the sphere.

**Figure 5.17:** Sphere problem setup.

For our particular interest, once the fluid particle spacing,  $\Delta x$  is set, the number of points to be placed on the sphere is determined in a similar fashion to the 2D case. Now the number of points is the quotient between the sphere surface area and the area given by the particle spacing, that is

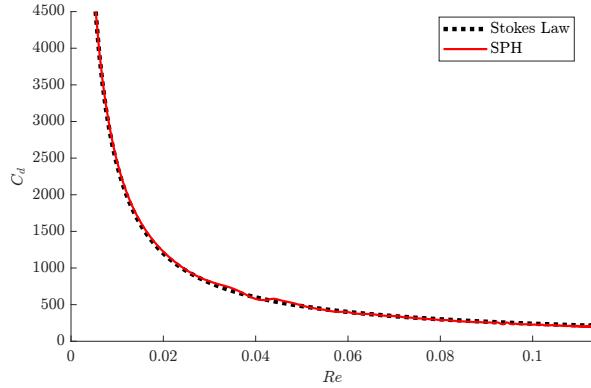
$$N = \lceil \frac{4\pi R^2}{\Delta x^2} \rceil.\tag{5.11}$$

This number of particles is chosen to achieve an area element of the particles  $\Delta s = \frac{4\pi R^2}{N} \approx \Delta x^2$ . We initialize the particles for a sphere of radius  $R = 0.5$ , placed at the centre of a rectangular box. The box has size  $L_x = 20$ ,  $L_y = 10$ ,  $L_z = 10$ , and it is filled with fluid particles initially at rest. The fluid phase has density  $\rho = 1$ ,  $p_B = p_0$  and  $\nu = 1$ . Periodic boundary conditions are imposed in all directions, and no-slip boundary conditions are set at the sphere surface. The flow is driven by a constant acceleration on the  $x$  direction,  $g_x = 5 \cdot 10^{-4} m/s^2$ . In Fig.5.17b we show the fluid and marker particles initialization for the sphere problem.

The Reynolds number is given by  $Re = \frac{\langle u_x \rangle D}{\nu}$ , where  $D$  is the diameter of the sphere and  $\langle u_x \rangle$  is the average velocity of the fluid particles in the  $x$  direction. The drag coefficient for the sphere is computed as

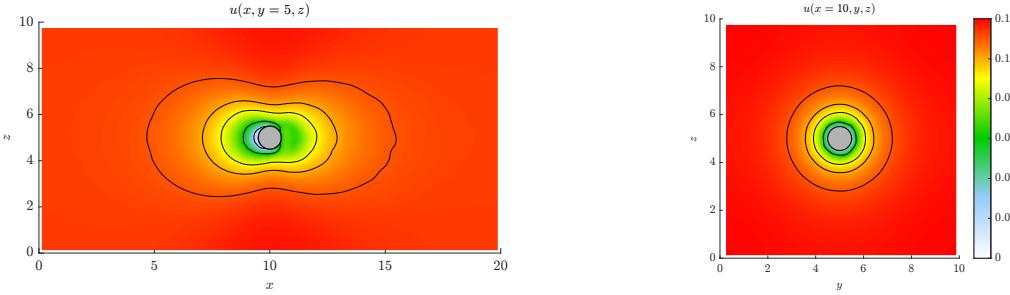
$$C_d = \frac{F_x}{\frac{1}{2}\rho\langle u_x \rangle^2 A}. \quad (5.12)$$

Where  $A$  is the cross-sectional area of the sphere,  $A = \pi R^2$ . At this low Reynolds regime, the drag coefficient follows Stokes law,  $C_d = 24/Re$ . Due to the low acceleration, the average speed varies slowly in time until steady state is reached. For each time instant we monitor the Reynolds number and the drag coefficient. In Fig.5.18 we show the drag coefficient as a function of the Reynolds number, compared with the theoretical Stokes law. It can be seen how the measured drag coefficient follows the theoretical curve, showing the expected decrease with the Reynolds number.

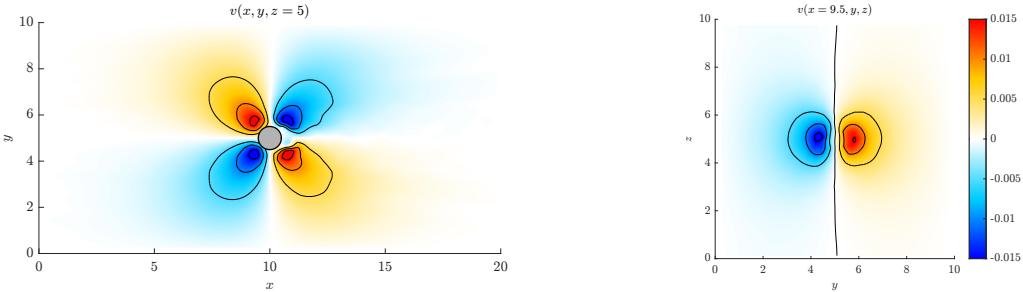


**Figure 5.18:** Drag coefficient of the sphere as function of the Reynolds number.

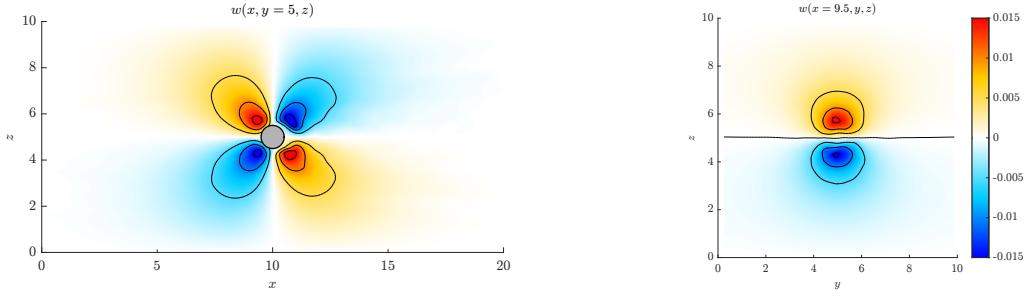
In Fig.5.19 to 5.21 we display the velocity field components around the sphere at longitudinal and transversal planes. The obtained velocity matches the expected low Reynolds laminar flow around a sphere.



**Figure 5.19:**  $u$  velocity around the sphere. Plane  $y = 5$  (left) and plane  $x = 10$  (right).



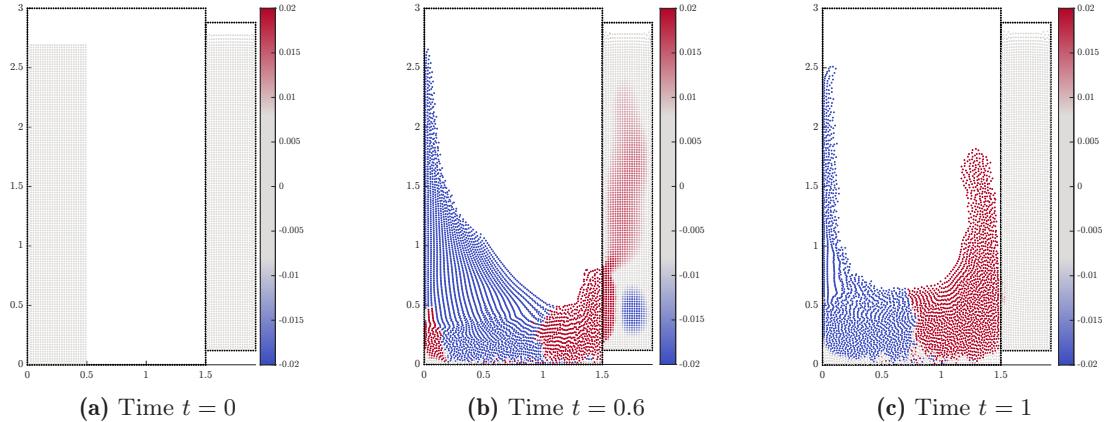
**Figure 5.20:**  $v$  velocity around the sphere. Plane  $z = 5$  (left) and plane  $x = 9.5$  (right).



**Figure 5.21:**  $w$  velocity around the sphere. Plane  $y = 5$  (left) and plane  $x = 9.5$  (right).

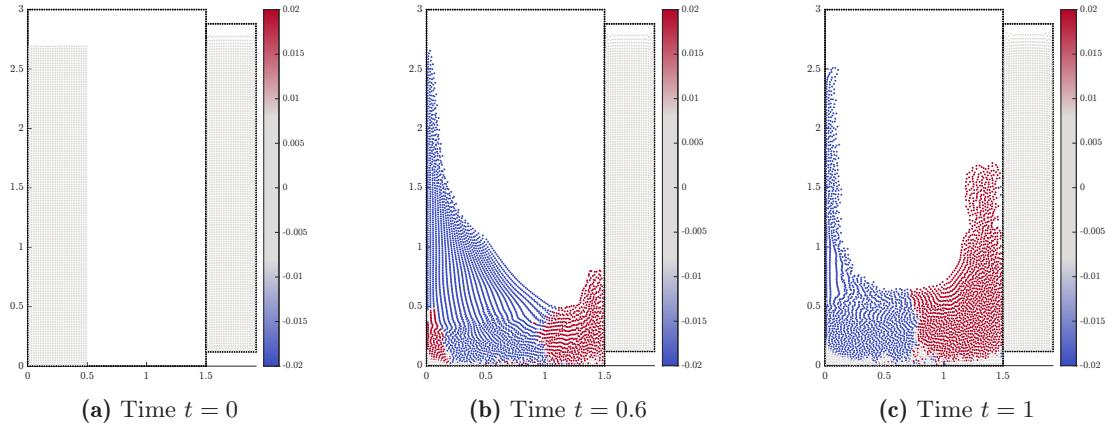
## 5.7 Dam Break with Adjacent Hydrostatic Tank

To show the importance of the interaction limiting for thin walls, we simulate a variation of the dam break problem. The setup of the problem is given by a typical dam break problem inside a rectangular box of dimensions  $L_x = 1.5$  and  $L_y = 3$ . Additionally, a rectangular fluid reservoir is placed adjacent to the box. Inside the main box, a fluid column of size  $0.5 \times 2.7$  is placed at rest. The fluid phase has density  $\rho = 1$ , and viscosity  $\nu = 0.1$ . The gravitational acceleration is set to  $\mathbf{g} = -10\hat{\mathbf{y}}$ . The right wall of the main box and the left wall of the fluid reservoir are separated by a very small distance of  $\Delta x = 0.01\Delta x$ . The interaction limiting described in Chapter 4 is used to prevent the fluid on each side from interacting with the wall of the other side. Otherwise this setup would not be feasible since the walls are too thin to be represented by this double layer of particles separated at such a small distance. This would cause the fluid to interact with the wall on the other side, and artificial repulsion would be seen on the fluid on both sides. The interaction limiting prevents this from happening, and the fluid on each side interacts only with the wall on its side. Moreover here we also demonstrate how the interaction limiting can be used to prevent fluid particles from either side to interact with each other, and only interact with the fluid on their side.



**Figure 5.22:** Dam break with adjacent hydrostatic tank. Interaction limiting of fluid particles disabled. Particles colored according to  $v$  velocity component.

Initially, the fluid column is frozen in time, until the fluid in the reservoir reaches hydrostatic equilibrium and remains at rest. After this is reached, the motion of the fluid column is released, and starts to spread along the box, until it impacts the wall and the fluid curves upwards and splashes back. In Fig.5.22 and Fig.5.23 we show the results of the simulation, when using or not the interaction limiting for fluid particles. It can be seen how when the wave impacts the wall, if no interaction limiter is used, the fluid on the hydrostatic tank feels the impact. This effect although unphysical might look rather small, but it is relevant enough to change the motion of the dam break wave. The interaction with fluid particles on the tank adds an increased friction, causing the wave to curve more and splash back more than when the interaction limiter is used. The interaction limiter prevents this unphysical effect from happening and the fluid on the tank remains at rest, as expected. And the simulation develops



**Figure 5.23:** Dam break with adjacent hydrostatic tank. Interaction limiting of fluid particles enabled. Particles colored according to  $v$  velocity component.

as if there was no tank on the side of the box.

## 5.8 Flow Driven by Rotating Polar Roses

As a last example we devise a problem with a convoluted geometry. This problem highlights the features of the method to represent complex geometries with marker particles. We simulate the fluid contained in between an outer epicycloid and an inner hypocycloid. The epicycloid is defined by the parametric equations

$$\begin{aligned} x &= \frac{R_1}{k} (k+1) \cos(\theta) - \frac{R_1}{k} \cos((k+1)\theta), \\ y &= \frac{R_1}{k} (k+1) \sin(\theta) - \frac{R_1}{k} \sin((k+1)\theta). \end{aligned} \quad (5.13)$$

And the hypocycloid is defined by the parametric equations

$$\begin{aligned} x &= \frac{R_2}{k} (k-1) \cos(\theta) + \frac{R_2}{k} \cos((k-1)\theta), \\ y &= \frac{R_2}{k} (k-1) \sin(\theta) - \frac{R_2}{k} \sin((k-1)\theta). \end{aligned} \quad (5.14)$$

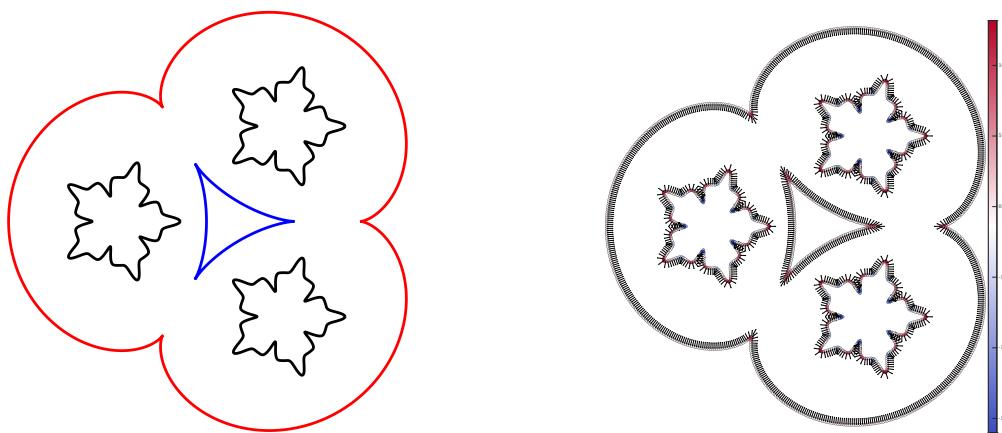
For  $\theta \in [0, 2\pi]$ . We set  $R_1 = 3$ ,  $R_2 = 1.5$  and  $k = 3$ . This choice makes the hypocycloid fit inside the epicycloid, and creates three regions in the domain, connected by a narrow strait formed by the sharp tips of the epicycloid and hypocycloid. In each of the three distinct regions we place a solid defined by a generalized polar rose curve, given by

$$\begin{aligned} r &= a(1+b\cos(k\theta))^m, \\ x &= r\cos(\theta), \\ y &= r\sin(\theta). \end{aligned} \quad (5.15)$$

We set the curve parameters to  $a = 1$ ,  $b = 0.4$ ,  $k = 5$  and  $m = 7$ . These solids are placed at the midpoint between the centre and the epicycloid maximum radius, i.e.  $\frac{5}{6}R_1$  from the centre. Their centres form angles  $0^\circ$ ,  $120^\circ$  and  $240^\circ$  with respect to the  $x$  axis. This curve generates a five point star like shape, with alternating tips and grooves. The shape presents high concave and convex curvatures, and sharp corners. The geometry of the problem is shown in Fig.5.24a, where the analytical curves are displayed. The discrete representation of the geometry with the marker particles is shown in Fig.5.24b, where the marker particles are colored according to the curvature, and their normal vectors are displayed.

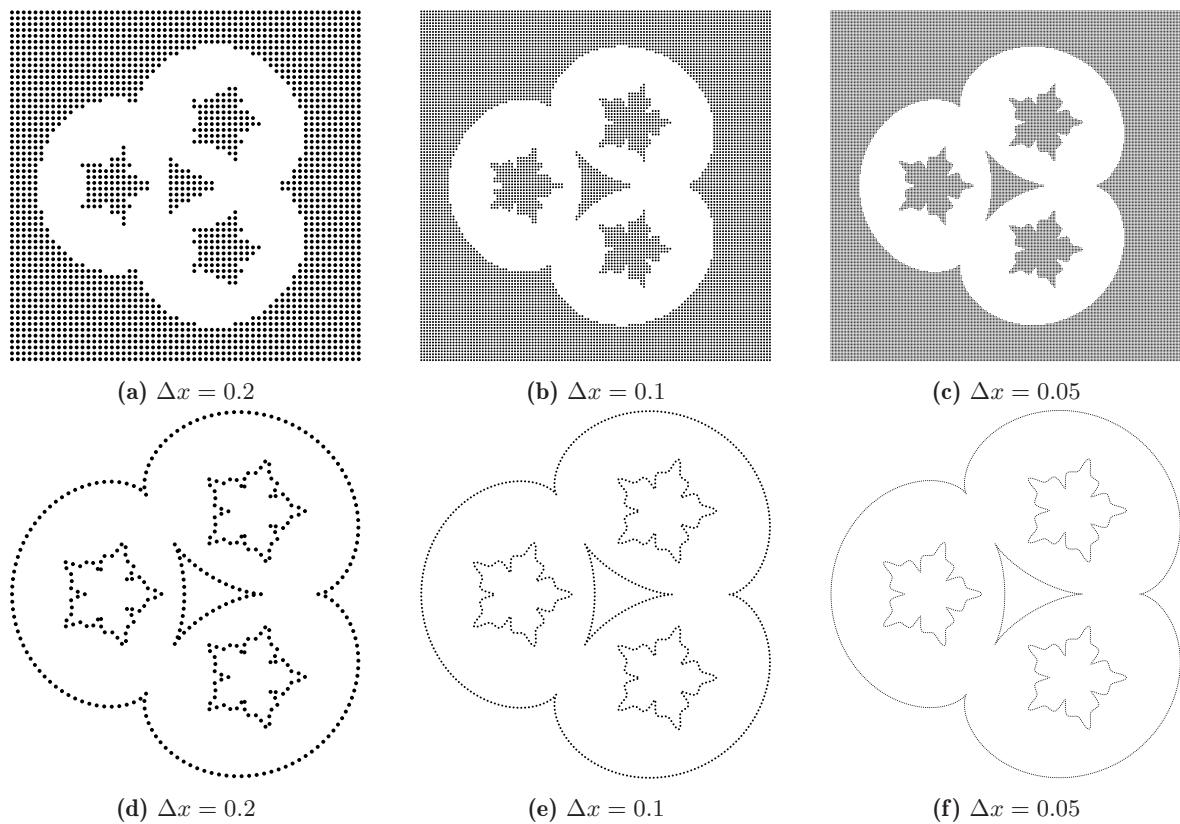
In Fig.5.25 we show the discretization of the solid walls using a cartesian lattice and the new method. The cartesian lattice discretization is shown in the top row, and the new method is shown in the bottom row. It can be seen how using a cartesian lattice, the geometry of the problem is not well represented for low resolutions, and high resolution is needed to capture the sharp corners and varying curvatures of the geometry. The new method, on the other hand, accurately represents the geometry of the problem for all resolutions, and increased resolution only serves to increase the accuracy of the SPH approximations.

The inside of the domain is filled with fluid particles with the following properties,  $\rho_0 = 1$ ,  $\nu = 1$ ,  $p_B = p_0$  and  $u_{\max} = 5$ . The fluid resolution is set to  $h = 0.05$ . The flow is driven by the rotation of the star solids with angular velocity  $\Omega = 3$ . The fluid particles in each region of the domain are tagged with a different value to visualize the flow patterns. In Fig.5.29 we show the time evolution of the flow in the domain. The particles are colored according to their tag value, and the flow patterns are shown as the fluid is advected by the rotating solids. The particles correctly move around the solid walls and bend along the sharp corners of the geometry, demonstrating the capability of the method to simulate the flow in complex geometries with sharp corners and varying curvatures.

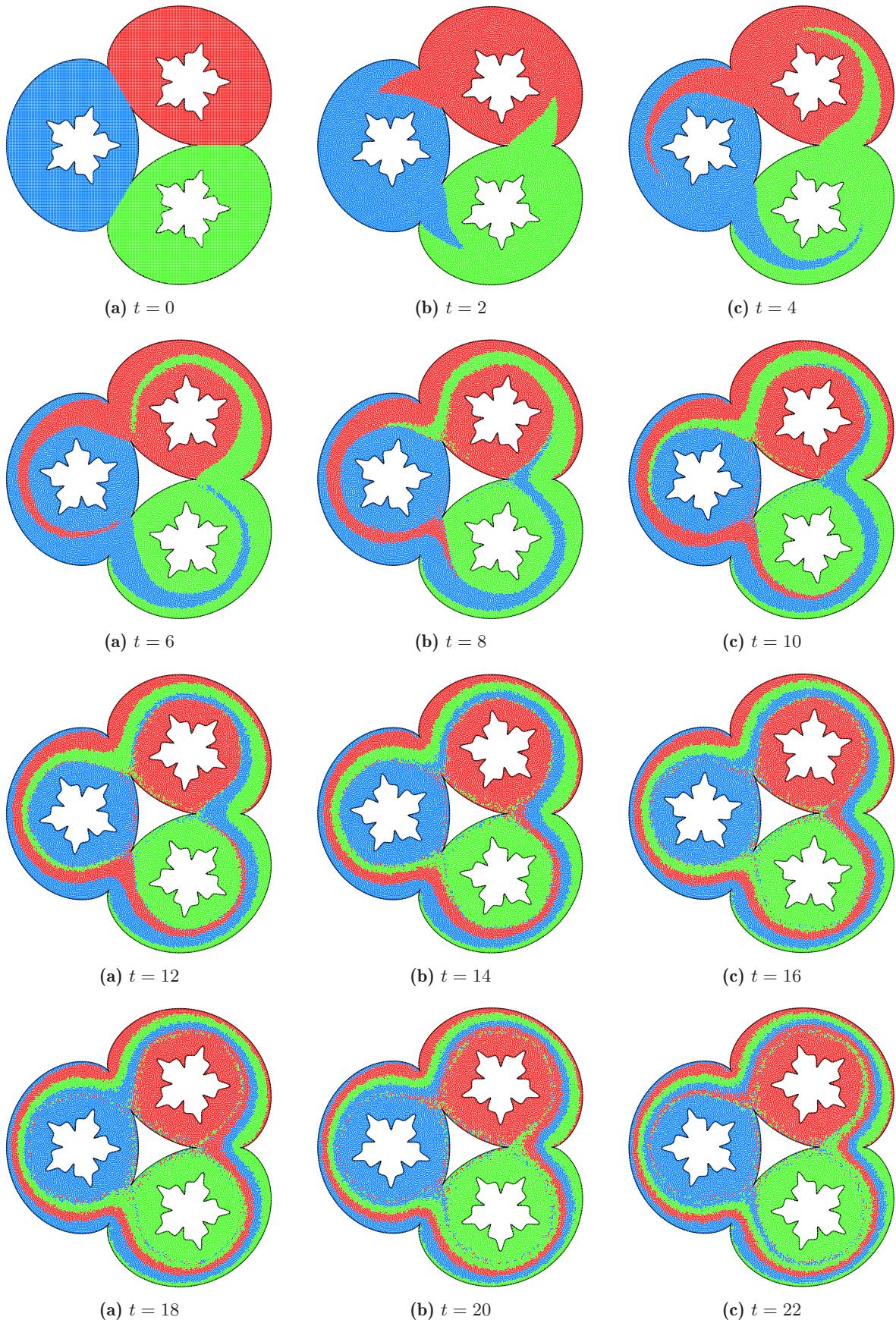


**(a)** Solid wall geometry, analytical curves. Epicycloid (red), hypocycloid (blue) and generalized polar rose curves (black). **(b)** Discrete particle representation of the solid walls, with normal vectors displayed. Particles colored according to curvature.

**Figure 5.24:** Geometry of the problem.



**Figure 5.25:** Discretization of the solid walls, using a cartesian lattice (top) and the new marker particle method (bottom).



**Figure 5.29:** Mixing of particles in the flow driven by the rotation of the polar rose shaped solids. Particles colored according to their tag value. Time evolution of the flow.

# Chapter 6

## Conclusions

This study presents an effective single-layer ghost particle boundary method for Smoothed Particle Hydrodynamics SPH, significantly improving the accuracy and flexibility of modeling fluid-solid interactions. The proposed approach focuses on accurate geometrical representation of solid boundaries. It introduces marker particles positioned along solid boundaries, which generate virtual particles that mirror key physical properties such as velocity and pressure to enforce the solid boundary conditions. The method is designed to provide a resolution-independent representation of complex solid geometries, enabling robust and reliable modeling even in scenarios with intricate boundary shapes. The single-layer formulation allows for the implementation of advanced techniques for handling sharp corners and thin solid profiles, areas that are traditionally problematic in particle-based simulations. These features are achieved without a significant increase in computational cost, making the method both accurate and efficient.

Validation of the proposed approach has been conducted through comparisons with analytical solutions and simulations of practical flow scenarios. These validations demonstrate the method's ability to replicate physical phenomena with high accuracy, confirming its robustness and versatility. Examples include the successful modeling of flows in geometrically complex domains with curved surfaces and sharp edges, such as the cylinder array and lattice, and the flow around an elliptical and triangular cylinder. The modelling of moving and accelerating solid boundaries has also been investigated in the context of the Taylor-Couette flow, the lid-driven cavity flow, and the flow around a moving square. The results show that the method can accurately capture the fluid-solid interactions in these scenarios, providing a reliable tool for a wide range of applications. Furthermore the formulation of the method allows for a simple and natural extension to three-dimensional simulations, as demonstrated by the successful implementation of the method in a three-dimensional flow scenario like the flow around a sphere.

The computational efficiency of the method is another notable feature. By streamlining the representation of solid boundaries through marker particles, the approach minimizes the computational overhead typically associated with more complex boundary treatments. This efficiency makes it suitable for large-scale simulations and applications where computational resources are limited.

Looking forward, there are several potential directions for future work. The simple interaction limiting for thin walls technique presented in this work could be further refined to generalize its applicability to a wider range of scenarios. Another area of potential study is the extension to fluid-structure-interaction problems, where the method could be used to model the interaction between fluid and deformable solid bodies.



## Appendix A

### Code

To implement the method described in this work, and run the simulations presented, we developed our own code using the OpenFPM library[15]. The code is written in C++/CUDA, and can both run using GPU acceleration, or in CPU only mode. The code uses the library features for particle parallel data structures, near neighbor lists, domain decomposition, dynamic load balancing, reduction operations, and other utilities. The source code can be found at [https://github.com/Jcela00/sph\\_bc](https://github.com/Jcela00/sph_bc).



# Bibliography

- [1] Adami, S., Hu, X., and Adams, N. “A generalized wall boundary condition for smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2012.05.005>.
- [2] Adami, S., Hu, X., and Adams, N. “A transport-velocity formulation for smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 241 (2013), pp. 292–307. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2013.01.043>.
- [3] Antuono, M., Pilloton, C., Colagrossi, A., and Durante, D. “Clone particles: A simplified technique to enforce solid boundary conditions in SPH”. In: *Computer Methods in Applied Mechanics and Engineering* 409 (2023), p. 115973. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2023.115973>.
- [4] Colagrossi, A. *SPHERIC Test Case 6: 2D Incompressible Flow Around a Moving Square Inside a Rectangular Box*. <https://www.spheric-sph.org/tests/test-06>.
- [5] Colicchio, G., Greco, M., and Faltinsen, O. “Fluid-body interaction on a Cartesian grid: dedicated studies for a CFD validation”. In: *Proc. IWWFB* 6 (2006).
- [6] DE, L., Le Touzé, D., and Alessandrini, B. “Normal flux method at the boundary for SPH”. In: *Proceedings of the 4th SPHERIC Workshop* (Jan. 2009).
- [7] DE, L., Le Touzé, D., and Alessandrini, B. “A modified no-slip condition in weakly-compressible SPH”. In: *Proceedings of the 6th International SPHERIC Workshop* (Jan. 2011).
- [8] Ellero, M. and Adams, N. “SPH simulations of flow around a periodic array of cylinders confined in a channel”. In: *International Journal for Numerical Methods in Engineering* 86 (May 2011), pp. 1027–1040. DOI: <10.1002/nme.3088>.
- [9] Ferrand, M., Laurence, D., and Rogers, B. “Improved Time Scheme Integration Approach for Dealing With Semi Analytical Wall Boundary Conditions in SPARTACUS2D”. In: (Jan. 2010).
- [10] Ghia, U., Ghia, K., and Shin, C. “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method”. In: *Journal of Computational Physics* 48.3 (1982), pp. 387–411. ISSN: 0021-9991. DOI: [https://doi.org/10.1016/0021-9991\(82\)90058-4](https://doi.org/10.1016/0021-9991(82)90058-4).
- [11] Gingold, R. A. and Monaghan, J. J. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389. ISSN: 0035-8711. DOI: <10.1093/mnras/181.3.375>.
- [12] González, Á. “Measurement of Areas on a Sphere Using Fibonacci and Latitude-Longitude Lattices”. In: *Mathematical Geosciences* 42.1 (Nov. 2009), pp. 49–64. ISSN: 1874-8953. DOI: <10.1007/s11004-009-9257-x>. URL: <http://dx.doi.org/10.1007/s11004-009-9257-x>.

- [13] Hu, X. and Adams, N. “A multi-phase SPH method for macroscopic and mesoscopic flows”. In: *Journal of Computational Physics* 213.2 (2006), pp. 844–861. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2005.09.001>.
- [14] Hu, X. and Adams, N. “A Constant-density Approach for Incompressible Multi-phase SPH”. In: *J. Comput. Physics* 228 (Apr. 2009), pp. 2082–2091. DOI: [10.1016/j.jcp.2008.11.027](https://doi.org/10.1016/j.jcp.2008.11.027).
- [15] Incardona, P., Leo, A., Zaluzhnyi, Y., Ramaswamy, R., and Sbalzarini, I. F. “OpenFPM: A scalable open framework for particle and particle-mesh codes on parallel computers”. In: *Computer Physics Communications* 241 (2019), pp. 155–177. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2019.03.007>.
- [16] Koshizuka, S., Nobe, A., and Oka, Y. “Numerical analysis of breaking waves using the moving particle semi-implicit method”. In: *International Journal for Numerical Methods in Fluids* 26.7 (1998). Cited by: 804, pp. 751–769. DOI: [10.1002/\(sici\)1097-0363\(19980415\)26:7<751::aid-fld671>3.0.co;2-c](https://doi.org/10.1002/(sici)1097-0363(19980415)26:7<751::aid-fld671>3.0.co;2-c).
- [17] Libersky, L. D. and Petschek, A. G. “Smooth particle hydrodynamics with strength of materials”. In: *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. Ed. by Trease, H. E., Fritts, M. F., and Crowley, W. P. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 248–257. ISBN: 978-3-540-46608-6.
- [18] Liu, A. W., Bornside, D. E., Armstrong, R. C., and Brown, R. A. “Viscoelastic flow of polymer solutions around a periodic, linear array of cylinders: comparisons of predictions for microstructure and flow fields”. In: *Journal of Non-Newtonian Fluid Mechanics* 77.3 (1998), pp. 153–190. ISSN: 0377-0257. DOI: [https://doi.org/10.1016/S0377-0257\(97\)00067-0](https://doi.org/10.1016/S0377-0257(97)00067-0).
- [19] Lucy, L. “A numerical approach to the testing of the fission hypothesis.” In: *The Astronomical Journal* 82 (1977), pp. 1013–1024. DOI: [10.1086/112164](https://doi.org/10.1086/112164).
- [20] Maciá, F., Antuono, M., González, L. M., and Colagrossi, A. “Theoretical Analysis of the No-Slip Boundary Condition Enforcement in SPH Methods”. In: *Progress of Theoretical Physics* 125.6 (June 2011), pp. 1091–1121. ISSN: 0033-068X. DOI: [10.1143/PTP.125.1091](https://doi.org/10.1143/PTP.125.1091).
- [21] Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touzé, D., and Graziani, G. “ $\delta$ -SPH model for simulating violent impact flows”. In: *Computer Methods in Applied Mechanics and Engineering* 200.13 (2011), pp. 1526–1542. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2010.12.016>.
- [22] Marrone, S., Colagrossi, A., Antuono, M., Colicchio, G., and Graziani, G. “An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers”. In: *Journal of Computational Physics* 245 (July 2013), pp. 456–475. DOI: [10.1016/j.jcp.2013.03.011](https://doi.org/10.1016/j.jcp.2013.03.011).
- [23] Monaghan, J. J. “Smoothed particle hydrodynamics”. In: *Reports on Progress in Physics* 68.8 (July 2005), p. 1703. DOI: [10.1088/0034-4885/68/8/R01](https://doi.org/10.1088/0034-4885/68/8/R01).
- [24] Monaghan, J. “Simulating Free Surface Flows with SPH”. In: *Journal of Computational Physics* 110.2 (1994), pp. 399–406. ISSN: 0021-9991. DOI: <https://doi.org/10.1006/jcph.1994.1034>.
- [25] Monaghan, J. “SPH without a Tensile Instability”. In: *Journal of Computational Physics* 159.2 (2000), pp. 290–311. ISSN: 0021-9991. DOI: <https://doi.org/10.1006/jcph.2000.6439>.

- [26] Morris, J. P., Fox, P. J., and Zhu, Y. “Modeling Low Reynolds Number Incompressible Flows Using SPH”. In: *Journal of Computational Physics* 136.1 (1997), pp. 214–226. ISSN: 0021-9991. DOI: <https://doi.org/10.1006/jcph.1997.5776>.
- [27] Negi, P. and Ramachandran, P. “How to train your solver: Verification of boundary conditions for smoothed particle hydrodynamics”. In: *Physics of Fluids* 34.11 (Nov. 2022), p. 117125. ISSN: 1070-6631. DOI: [10.1063/5.0126234](https://doi.org/10.1063/5.0126234). URL: <https://doi.org/10.1063/5.0126234>.
- [28] Parmar, M., Haselbacher, A., and Balachandar, S. “On the unsteady inviscid force on cylinders and spheres in subcritical compressible flow”. In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 366 (Mar. 2008), pp. 2161–75. DOI: [10.1098/rsta.2008.0027](https://doi.org/10.1098/rsta.2008.0027).
- [29] Quinlan, N. J., Basa, M., and Lastiwka, M. “Truncation error in mesh-free particle methods”. In: *International Journal for Numerical Methods in Engineering* 66.13 (2006), pp. 2064–2085. DOI: <https://doi.org/10.1002/nme.1617>.
- [30] Ricci, F. “Variable Resolution Smoothed Particle Hydrodynamics Schemes for 2-D and 3-D Viscous Flows”. Ph.D. Dissertation. Dissertation. New Jersey Institute of Technology, Aug. 2023. URL: <https://digitalcommons.njit.edu/dissertations/1685>.
- [31] Shadloo, M. S., Zainali, A., Sadek, S. H., and Yildiz, M. “Improved Incompressible Smoothed Particle Hydrodynamics method for simulating flow around bluff bodies”. In: *Computer Methods in Applied Mechanics and Engineering* 200.9 (2011), pp. 1008–1020. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2010.12.002>.
- [32] Shao, S. and Lo, E. Y. “Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface”. In: *Advances in Water Resources* 26.7 (2003), pp. 787–800. ISSN: 0309-1708. DOI: [https://doi.org/10.1016/S0309-1708\(03\)00030-7](https://doi.org/10.1016/S0309-1708(03)00030-7).
- [33] Takeda, H., Miyama, S. M., and Sekiya, M. “Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics”. In: *Progress of Theoretical Physics* 92.5 (Nov. 1994), pp. 939–960. ISSN: 0033-068X. DOI: [10.1143/ptp/92.5.939](https://doi.org/10.1143/ptp/92.5.939). URL: <https://doi.org/10.1143/ptp/92.5.939>.
- [34] Wang, Z.-B., Chen, R., Wang, H., Liao, Q., Zhu, X., and Li, S.-Z. “An overview of smoothed particle hydrodynamics for simulating multiphase flow”. In: *Applied Mathematical Modelling* 40.23 (2016), pp. 9625–9655. ISSN: 0307-904X. DOI: <https://doi.org/10.1016/j.apm.2016.06.030>.



# Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated.  
I have only used the resources given in the list of references.

Garching, December 15, 2024



---

(Signature)