Eleonora Spricigo, MSc. Eng.

Institute of Fluid-Flow Machinery
Polish Academy of Sciences

Application of Smoothed Particle Hydrodynamics method to interfacial two-phase flows

Zastosowanie metody hydrodynamiki cząstek wygładzonych (SPH) do przepływów z powierzchniami międzyfazowymi

Doctoral dissertation
submitted to the Scientific Board of
Institute of Fluid-Flow Machinery
Polish Academy of Sciences

PhD advisor:
Jacek Pozorski, Prof. D.Sc. Eng.

Gdańsk, April 2024

The study has been carried out in the framework ITN-EID project COMETE No. 813948, funded by EU H2020 Maria Skłodowska-Curie Actions.

I would like to express the deepest gratitude to my advisor Professor Jacek Pozorski for giving me the opportunity to work on this project, and for his unlimited support and valuable discussion. Special thanks to the coordinator, Professor Cristian Marchioli, who took the time to organise the COMETE project efficiently. I would also like to acknowledge Professors Carlo Poloni, Enrico Nobile, and the entire team from my secondment in ESTECO SpA for the wonderful collaboration. Thanks should go to my colleagues Michał Olejnik, Adam Kajzer, Michał Rajek and Arkadiusz Grucelski for the most fruitful discussions during coffee breaks and for their precious friendship. I would like to thank Justyna Przesmycka from the Human Resources department, who kindly helped me through various administrative issues. I must thank the friends that made my time in Gdańsk memorable: Thanushree, Burcuk, Vladyslava, and Anastasia.

Un enorme ringraziamento ai miei genitori Giuseppe e Margherita per avermi insegnato tutto quello che conosco, soprattutto la tenacia. Grazie a mio fratello Sebastiano, che è il mio migliore amico. Un ringraziamento anche a zia Paola che mi ha motivato (nonchè ospitata e nutrita) durante la stesura di questa tesi. Grazie alla nonna Carla e a tutto il resto della mia splendida famiglia per il loro costante e amorevole sostegno. Grazie ai miei fantastici amici Gianluca ed Eleonora, che mi sono stati vicini anche quando ci separavano chilometri di terra e acqua. Dulcis in fundo ringrazio Alberto, che ha creduto in me e nelle mie capacità sin dal primo giorno: *alla nostra libertà emotiva e ad una vita magnifica*.

Abstract

Studying multiphase flows is crucial for gaining a better understanding and advancing current technology in various industrial processes and everyday life. In the realm of contemporary physics and engineering, numerical simulations play a vital role in aiding designers and technicians in this endeavour.

The Smoothed Particle Hydrodynamics (SPH) method employs particle-based discretisation, where each particle represents fluid properties and moves within the computational domain, mirroring the behaviour of the fluid. Interactions between particles are modelled using a smooth interpolating function, called kernel. In the field of Computational Fluid Dynamics (CFD), there exist various Eulerian approaches capable of solving multiphase flows. As a Lagrangian alternative, SPH has revealed to be a convenient option for selected classes of multiphase flows as it does not necessitate any specific measures to handle interface dynamics. Nevertheless, attention must be given to proper treatment of fluid density and viscosity changes across the interface, especially when the density and viscosity ratios are elevated.

Motivated by limited research on multiphase viscous flows in the current SPH literature, our aim is to assess the effectiveness of the model and the functionalities of the in-house code in managing these complex simulations. One of the goals is to define a proper expression for the viscous force within weakly compressible SPH (whose acronym is WCSPH). Hence, in this dissertation we propose a modified Violeau, Español and Revenga (mVER) formulation that is in accordance with the physical viscous force term in presence of the interface by implementing a suitable smoothing.

The new formulation is first validated against the steady state, single- and two-phase Poiseuille flow of known analytical solutions. An additional advantage of the Poiseuille flow benchmark is its physical simplicity, allowing us to focus on the assessment of the mVER formula. Since the new formulation has succeeded the first step, we considered the second benchmark, often encountered in the literature on multiphase flows, which is a gas bubble rising in liquid. In this study are considered two regimes of the bubble rise; they differ on the basis of the shape taken by the bubble when subjected to buoyancy and the shear effect, including at the interface. In the first regime the two fluid phases have relatively small difference in density and viscosity values, and the final bubble shape is ellipsoidal. In the second regime the density and viscosity ratios are much higher; consequently, the bubble is strongly deformed and assumes a so-called skirt shape. The first case, the ellipsoidal regime, has proved to be less challenging and we obtained a good agreement with the reference data in terms of the bubble rising velocity and final bubble shape, also measured using the circularity parameter. Utilising SPH to simulate the second regime, we have struggled to obtain the elongated skirt that was expected. As a positive outcome, the top part of the bubble had a shape comparable to the reference one. Moreover, the resulting rising velocity and the circularity parameters were close to the benchmark. Both cases have however showed an oscillatory behaviour of the rising velocity; this is a well-known issue that occurs in weakly compressible approaches.

The mVER formulation is also tested against more complex applications, including the sloshing phenomenon and the moon pool, a key feature in drill-ships and offshore platforms. It is an opening through the hull of vessels that allows underwater activity from on-board. Since the sloshing phenomenon has not been simulated before in our code, the sloshing tank benchmark is a natural preparatory study for the moon pool. Comparing our results with the reference analytical formulae stemming from the potential flow theory, we obtained good agreement for the wave height, while the sloshing force had a lower amplitude, because it manifested an inevitably viscous behaviour.

As the last application we studied the moon pool. Its presence influences the stability of ship motion, necessitating a thorough investigation, particularly for vessels like drill-ships frequently subjected to harsh open-sea conditions. Despite significant advancements in experimental analysis, CFD emerges as a viable and relatively convenient alternative, in some cases eliminating the need for complex and expensive towing tank laboratory studies. Considering this aspect, we have selected a recent reference in the literature, both experimental and computational, and implemented the same case in our SPH code. Notwithstanding the necessary simplifications, both in geometry (e.g., our moon pool is two-dimensional) and in the physical model (e.g., assuming periodic, rather than open, boundary conditions), the obtained dominant frequency of the sloshing flow inside the moon pool was comparable with the benchmark.

Still regarding the moon pool, we have also included a study on the recess which gives the moon pool the characteristic L-shape. Recently, the recess has demonstrated to be useful for assembling and putting in water tools and devices, particularly on drill-ships. Additionally, the recess affects the wave dynamics within the moon pool. We delve into the optimisation of the recess geometry, aiming to find the best solution, or a set of solutions, that minimise the wave height inside the moon pool. We have conducted the optimisation by coupling the SPH code with the software modeFRONTIER, provided by ESTECO SpA, and testing various flow conditions. Utilising the simplified moon pool already tested, we have firstly run a set of initial simulations by varying the recess height and length, as well as the frequency of the surface waves. Out of these simulations we can create a new predictive mathematical model using the response surface methodology. We explored this meta-model to compare as many new designs of the recess as we need to find the one (or ones) that performs best under the given flow conditions.

In the author's opinion, possible directions for future development are: an investigation, in the WCSPH framework, of the relationship between the smoothing kernel expression and the observed oscillations of the solution; a further validation of the mVER formulation for new benchmarks, also applying other SPH software; a research on a physically sound and computationally efficient implementation of open boundary conditions.

Streszczenie: Badanie przepływów wielofazowych ma istotne znaczenie dla lepszego zrozumienia zjawisk oraz doskonalenia rozwiązań technicznych stosowanych w różnych procesach przemysłowych i w życiu codziennym. We współczesnej fizyce i technice symulacje numeryczne odgrywają ważną rolę, pomagając pracownikom naukowym i inżynierom w tym przedsięwzięciu.

Metoda hydrodynamiki cząstek wygładzonych (ang. *Smoothed Particle Hydrodynamics*, SPH) wykorzystuje dyskretyzację opartą na koncepcji cząstek: każda cząstka reprezentuje właściwości płynu i porusza się w obszarze obliczeniowym, odzwierciedlając zachowanie płynu. Interakcje pomiędzy cząstkami modeluje się za pomocą gładkiej funkcji interpolującej zwanej jądrem. W obliczeniowej mechanice płynów (ang. *Computational Fluid Dynamics*, CFD) istnieją różne metody wykorzystujące podejście Eulera umożliwiające wyznaczanie przepływów wielofazowych. Jako alternatywa korzystająca z podejścia Lagrange'a metoda SPH okazała się dogodną opcją dla wybranych klas przepływów wielofazowych, ponieważ nie wymaga specjalnych sposobów wyznaczania dynamiki powierzchni międzyfazowej (powierzchni rozdziału faz, interfejsu). Niemniej jednak należy zwrócić uwagę na sposób ujęcia zmian gęstości i lepkości płynu na granicy faz, zwłaszcza gdy zmiany te są znaczne.

Ponieważ prace na temat modelowania wielofazowych przepływów lepkich stanowią relatywnie niewielką część aktualnej literatury z zakresu metody SPH, naszym celem jest ocena modelu obliczeniowego oraz funkcjonalności kodu numerycznego wykorzystywanego i rozwijanego w kontekście tych złożonych symulacji. Jednym z celów jest zdefiniowanie prawidłowego wyrażenia siły lepkości w słabo ściśliwym podejściu SPH, określanym akronimem WCSPH (ang. *weakly compressible*, WC). W niniejszej rozprawie proponujemy zmodyfikowaną formułę Violeau, Español i Revenga (mVER), która jest zgodna z fizykalnym członem sił lepkości w obecności granicy faz poprzez zastosowanie odpowiedniego wygładzenia.

Nowa formuła została zweryfikowana w przypadku przepływu Poiseuille'a w stanie ustalonym w wariantach jedno- i dwufazowych, których rozwiązania analityczne są znane. Dodatkową zaletą przepływu Poiseuille'a jako problemu wzorcowego (ang. *benchmark*) jest fizykalna prostota, co pozwoliło skupić się na ocenie formuły mVER. Ponieważ przeszła ona pomyślnie pierwszy etap, rozważono inny często spotykany w literaturze problem wzorcowy dla przepływów z powierzchnią międzyfazową: wznoszenie się pęcherzyka gazu w cieczy. W tym zagadnieniu rozważa się dwa reżimy ruchu; różnią się one kształtem, jaki przyjmuje pęcherzyk pod wpływem siły wyporu i efektu sił ścinania, w tym na granicy rozdziału faz. W pierwszym reżimie dwie fazy płynne cechuje stosunkowo niewielka różnica w wartościach gęstości i lepkości, a ostateczny kształt pęcherzyka jest elipsoidalny. W drugim reżimie stosunki gęstości i lepkości są znacznie wyższe; w efekcie pęcherzyk ulega silnej deformacji i przyjmuje kształt podobny do dzwonu lub spódnicy (ang. *skirted regime*). Pierwszy przypadek (reżim elipsoidalny) okazał się mniej wymagający obliczeniowo, co pozwoliło uzyskać dobrą zgodność z danymi referencyjnymi dla prędkości wznoszenia się pęcherzyka i jego końcowego kształtu, określanego również za pomocą parametru kolistości. Stosując SPH w celu symulacji drugiego reżimu, napotkaliśmy na trudności w uzyskaniu oczekiwanego kształtu pęcherzyka. Pozytywnym rezultatem było to, że jego górna część uzyskała kształt porównywalny do znanego z literatury, a końcowa prędkość wznoszenia pęcherzyka oraz parametr kolistości były zbliżone do referencyjnych.

Obydwa przypadki wykazały jednak oscylacyjny przebieg prędkości wznoszenia; jest to skądinąd znany problem występujący w podejściach słabo-ściśliwych.

Sformułowanie mVER testowano także pod kątem bardziej złożonych zastosowań, w tym zjawiska ruchu cieczy w zbiorniku z powierzchnią swobodną (ang. *sloshing* – chlupotanie, rozkołys) oraz w specyficznej konfiguracji tak zwanego basenu księżycowego (ang. *moon pool*), występującego na statkach wiertniczych i platformach morskich. Jest to otwór w kadłubie statku ułatwiający prowadzenie prac podwodnych z pokładu. Ponieważ w naszym kodzie nie symulowano wcześniej zagadnienia chlupotania, przypadek wzorcowy ruchu cieczy w zbiorniku stanowił naturalny wstęp do symulacji SPH basenu księżycowego. Porównując uzyskane wyniki z formułami analitycznymi wynikającymi z teorii przepływu potencjalnego, uzyskaliśmy dobrą zgodność wysokości fali, natomiast zmienna w czasie siła oddziaływania na ściany zbiornika miała mniejszą amplitudę z uwagi na efekty sił lepkości.

Jako ostatni w rozprawie przypadek przepływu wielofazowego rozważono basen księżycowy. Jego obecność wpływa na stabilność ruchu statku, co wymaga dokładnych badań, szczególnie w przypadku jednostek takich jak statki wiertnicze często poddawanych trudnym warunkom na otwartym morzu. Pomimo znacznych postępów w zakresie eksperymentu, CFD okazuje się realną i stosunkowo dogodną alternatywą, w niektórych przypadkach eliminując potrzebę złożonych i kosztownych badań na basenie modelowym. Biorąc pod uwagę ten aspekt, wybraliśmy najnowsze dostępne z literaturze studium, zarówno eksperymentalne jak i obliczeniowe, i zaimplementowaliśmy ten sam przypadek w kodzie SPH. Niezależnie od niezbędnych uproszczeń, zarówno geometrii (np. nasz basen księżycowy jest dwuwymiarowy), jak i modelu fizycznego (np. przyjęcie okresowych, zamiast otwartych, warunków brzegowych), uzyskano dobrą zgodność dominującej częstotliwości fal powierzchniowych.

Ponadto dla przypadku basenu księżycowego podjęto analizę rozmiaru wnęki (ang. *recess*), która nadaje basenowi charakterystyczny kształt litery L poprzez wypłycenie jego części. W ostatnim czasie takie rozwiązanie okazało się użyteczne w praktyce morskiej, szczególnie na statkach wiertniczych, do montażu urządzeń podwodnych. Dodatkowo wnęka wpływa na dynamikę fal powierzchniowych w tym basenie. Przeprowadziliśmy optymalizację geometrii wnęki, mając na celu znalezienie najlepszego rozwiązania lub zestawu rozwiązań, które minimalizują wysokość fali wewnątrz basenu księżycowego. Optymalizację wykonaliśmy sprzęgając kod SPH z oprogramowaniem modeFRONTIER dostarczonym przez ESTECO SpA i testując różne warunki przepływu. Wykorzystując badany wcześniej uproszczony basen księżycowy, przeprowadziliśmy najpierw szereg wstępnych symulacji, zmieniając wysokość i długość wnęki, a także częstotliwość fal powierzchniowych. Na tej podstawie, wykorzystując metodę powierzchni odpowiedzi, stworzyliśmy nowy predykcyjny model matematyczny. Zastosowaliśmy ten metamodel, aby porównać odpowiednią liczbę konfiguracji geometrycznych wnęki i znaleźć tę (lub te), która jest najlepsza w danych warunkach przepływu.

Zdaniem autorki, niektóre możliwe kierunki dalszych prac to: zbadanie, w ramach podejścia WCSPH, związku pomiędzy jądrem interpolacji a obserwowanymi oscylacjami rozwiązania; dalsza weryfikacja formuły mVER w testach porównawczych, także z zastosowaniem innego oprogramowania SPH; badania w zakresie poprawnej fizykalnie i wydajnej obliczeniowo implementacji otwartych warunków brzegowych.

Contents

List of Figures

List of Tables

List of Symbols

Acronyms

$\delta-$SPH	diffusive term added to continuity equation in SPH
AKR	Anisotropic Kriging method
ANOVA	Analysis of Variance
ALE	Arbitrary Lagrangian-Eulerian
BC	Boundary Conditions
BEM	Boundary Element Method
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CIP	Constrained Interpolation Profile
CPU	Central Processing Unit
CSF	Continuum Surface Force
CUDA	Compute Unified Device Architecture
DACE	Design and Analysis of Computer Experiments
DBC	Dynamic Boundary Conditions
DOE	Design of Experiments
EoS	Equation of State
FF	Full Factorials
FFT	Fast Fourier Transform
FT	Front-Tracking
GPBC	Fixed Ghost Particles Boundary Conditions
GP	Gaussian Processes
GPU	Graphics Processing Unit
KN	Shepard K-Nearest
KR	Kriging method
HOCT	High-Order Core Triangle kernel
ISPH	Truly Incompressible Smoothed Particle Hydrodynamics

LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
LGA	Lattice Gas Automation
LS	Level Set
MAC	Marker-and-Cell
mF	modeFRONTIER
MLS	Moving Least Squares
MMPS	Multiphase Moving-Particle Semi-Implicit
MPS	Moving-Particle Semi-Implicit
NN	Neural Network
OpenCL	Open Computing Language
PLIC	Piecewise Linear Interface Calculation
PFM	Phase-Field Method
PPE	Pressure Poisson Equation
RANS	Reynolds Averaged Navier-Stokes
RBF	Radial Basis Function
RSM	Response surface Methodology
SLIC	Simple Line Interface Calculation
SM	Streaming Multiprocessor
SPH	Smoothed Particle Hydrodynamics
SS-ANOVA	Smoothing Spline ANOVA
SVD	Singular Value Decomposition
VOF	Volume of Fluid
WCSPH	Weakly Compressible Smoothed Particle Hydrodynamics
XSPH	artificial shifting of SPH particles to improve their distribution

Indexes

0	reference value
$antisym$	antisymmetric
$Arith$	Arithmetic
AV	Artificial Viscosity formulation
a, b	neighbouring particles a and b
α, β	exponents
b	body forces
i, j	coordinates in Einstein notation
HA	Hu and Adams formulation
$Harm$	Harmonic
k, l	phase indicators
M	Morris formulation
$neigh$	neighbours
v	viscous
sym	symmetric
s	sloshing
st	surface tension
t	body forces
w	wall

Greek letters

α	artificial viscosity dimensionless coefficient
γ	heat capacity ratio
Γ	boundary of the domain
δ_{ij}	Kronecker delta
$\delta(\mathbf{r})$	unit impulse
δt	time step
ϵ	interface sharpness correction coefficient
ζ	bulk viscosity

ζ_s	sloshing wave height
θ	particle number density
κ	local curvature
λ	second viscosity
μ	dynamic viscosity
ν	kinematic viscosity
Ξ	interface correction
ρ	fluid density
$\sigma_{\mathbf{r}}$	standard deviation
σ_{st}	surface tension coefficient
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\boldsymbol{\tau}$	viscous stress tensor
Ω	domain volume
$\Omega_{\mathbf{r}}$	compact support

Roman letters

$A(\mathbf{r})$	arbitrary scalar field
$\mathbf{A}(\mathbf{r})$	arbitrary vector field
B	kernel support radius
c	colour function
$\tilde{c}$	smoothed colour function
C'	kernel normalisation constant
$C^{2,4,6}$	differentiability class of the kernel
d	spatial dimension
d_{hull}	hull draft
D_{hull}	hull depth
$\mathbf{e}$	unit vector of relative particle position
E	total energy
Eo	Eötvös number
$\mathbf{f}$	force

$\mathrm{f}_{\mathrm{b},x}$	rectilinear sinusoidal sloshing force
f	frequency
f_ω	sloshing frequency
h	smoothing length
H	characteristic height
$\mathscr{H}$	Hamiltonian
$H_{wave}^{P_i}$	wave height for at the position of the i-th probe
$\mathbf{I}$	identity tensor
$\mathscr{I}$	action of a mechanical system
$J_{\mathbf{A}}$	diffusion coefficient
$\mathbf{k}$	wave vector
L	characteristic length
L_{mp}	hull depth
$\mathscr{L}$	Lagrangian
m	particle mass
m	sloshing amplitude
$\hat{\mathbf{n}}$	unit vector normal to the interface
N	number of particles
P	pressure
p	canonical momentum
P_i	probe $i = 1, 2, 3$
q	normalised smoothing length
q_i	generalised position of the i-th particle
$\dot{q}_i$	generalised velocity of the i-th particle
$\mathbf{r}$	vector of particle position
Re	Reynolds number
R_H	recess height
R_L	recess length
RMS_{err}	Root Mean Square error
s	artificial speed of sound

S	entropy
t	time
T	temperature
U	internal (thermal) energy
U_{kin}	kinetic energy
U_{pot}	potential energy
$\mathbf{v}$	vector of velocity
V	particle volume
v_{max}	maximum velocity
W	kernel function
x, y	Cartesian coordinates

1 Introduction

This manuscript focuses on multiphase flows; for those unfamiliar with the term, these flows are a combination of the phases solid, liquid, and gas. However, we will not consider miscible flows, where components are mixed at the molecular level, but rather focus on immiscible flows, where fluids are separated by an interphasial surface called interface. Generally, the multiphase flows can be categorized according to their physical nature: solid-liquid, solid-gas, liquid-gas and liquid-liquid, and eventually, the three of them together, such as solid-liquid-gas. In cases involving a solid phase, the interface is shape-retaining, while in liquid-gas or liquid-liquid scenarios, the interface can be highly deformable.

Multiphase flows are prevalent in everyday life and industrial processes. Examples in nature include sandstorms, rain droplets, soil erosion, and ocean waves; while industrial applications encompass fluidized beds, water jet cutting, boilers, and airlift pumps.

Given the common occurrence of multiphase flows, the scientific community is motivated to study these phenomena to better understand of their physics as well as current technological advancements. Computational Fluid Dynamics (CFD) facilitates this work for designers and technicians. The power of numerical analysis relies simply on the need of a computing machine rather than an entire experimental facility, which for some tests may be extremely expensive. Nevertheless, CFD modelling can require non-trivial capabilities and computing power which sometimes place limitations.

In fluid dynamics there are two main points of view called Eulerian and Lagrangian approaches. In general, the Eulerian approach is a way to observe the flow behaviour from a fixed point of view, and the functions featuring the flow may be time-dependent. On the other hand, the Lagrangian approach examines fluid motion in terms of trajectories of fluid elements (often called fluid particles). Hence, the CFD methods are mainly classified as Lagrangian or Eulerian.

Within the Lagrangian methods, the Smoothed Particle Hydrodynamics (SPH) stands out as a mesh-free approach. SPH discretises the flow domain using particles, each representing a point with associated mass, phase indicator, and other material properties. The interface between different phases becomes easily detectable as the surface between sets of particles belonging to distinct phases. Therefore, SPH proves well-suited for simulating multiphase flows.

Our team has been dedicated to studying multiphase flows and particle methods for over two decades. Over the last decade, an in-house code has been developed and enhanced for multiphase flows. This manuscript

represents our most recent efforts; we have upgraded the code to incorporate improvements to the SPH physical model and new related benchmarks.

The dissertation is organized as follows. The current part, Chapter 1 introduces the context of the research topic, referring to the most common CFD methods, acknowledges the SPH state of the art and reveals our research's aim and motivation. Chapter 2 delves into the details of the theoretical SPH background, so it will discuss the interpolation method, define the SPH formalism, and show how the governing equations are approximated in SPH. In Chapter 3 we present the novel formulation for the viscous force term, called the modified Violeau, Español and Revenga (mVER) formulation, validating it through the steady-state Poiseuille flow for single- and two-fluid scenarios and the rising bubble benchmark. More complex problems are treated in Chapter 4, the sloshing tank and the moon pool. Firstly, we show the results obtained performing the sloshing tank, a natural benchmark that serves to model the moon pool. We attempt a moon pool design optimisation in Chapter 5 coupling our code with the software modeFRONTIER, provided by ESTECO SpA. The chapter also includes an overview of the optimisation method. Conclusive thoughts about our achievements come in Chapter 6 together with future work suggestions. The manuscript concludes with Appendix which concerns the code information, specifically the parallelization.

1.1 Numerical methods for multiphase flows

Ancient civilizations possessed practical knowledge of fluid mechanics until after the 17th century, when the fundamental theoretical insights into the macroscopic behaviour of fluids gradually started to develop. In that period, renowned figures such as Archimedes, Johann Bernoulli, Leonhard Euler, Claude-Louis Navier, George Gabriel Stokes, Ludwig Prandtl and others made significant contributions. The foundation of modern fluid dynamics was laid in the 20th century. Valuable theoretical insights can be found in the works such as the book by Landau and Lifshitz [99], the textbook by Cahn and Hilliard [24] and the guide on fluid dynamics by Batchelor [9]; all of them cover the basics of interfacial and multiphase flows.

Pioneering CFD calculations were conducted in the 1940s, but the first work using computer took place in the late 1950s at Los Alamos National Lab. Led by Francis H. Harlow, a prominent figure in CFD development, the T3 research group played a crucial role. In the realm of numerical methods for multiphase flows, there are multiple approaches to explore; therefore, creating an exhaustive overview of all existing methods poses a considerable challenge. Nevertheless, studies as the ones by Mirjalili et al. [120], Soligo et al. [174], Olejnik [140], and books by Prosperetti and Tryggvason [155], Tryggvason et al. [189] and Brennen [20] offer valuable

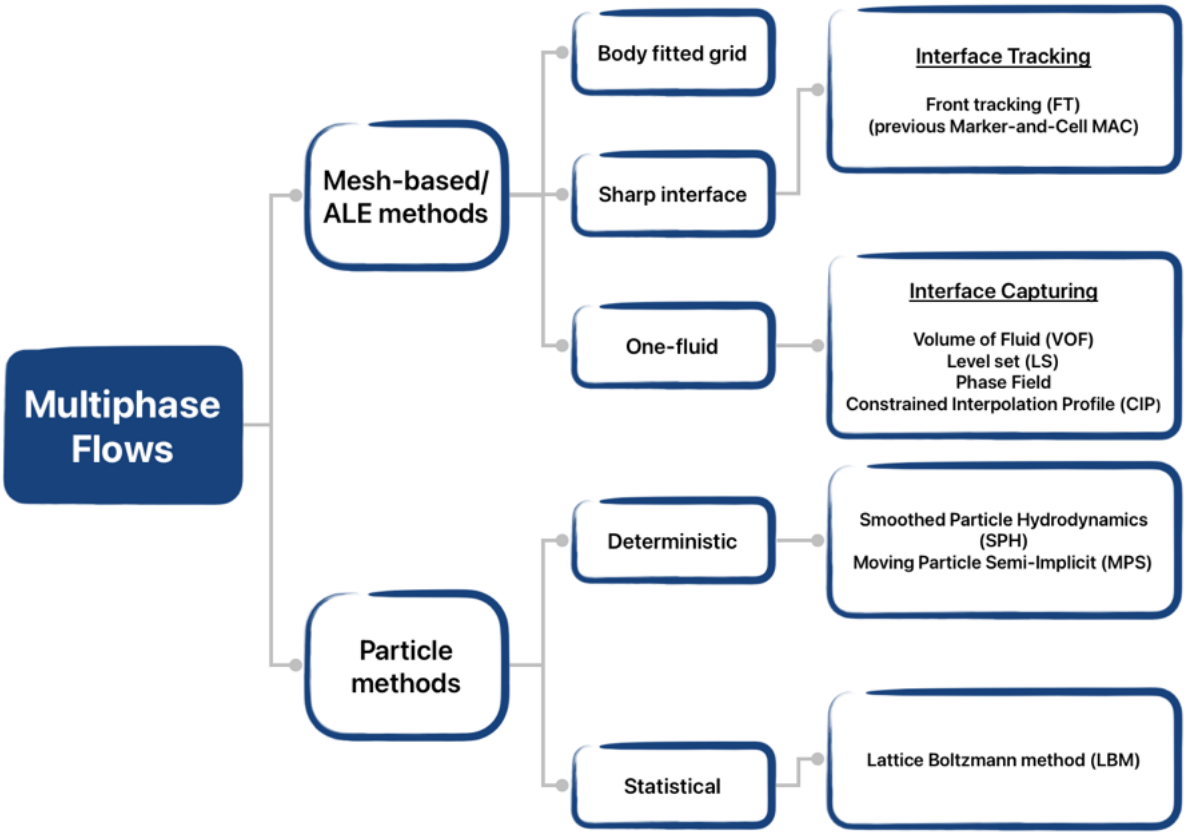


Figure 1.1: Classification of the most common computational approaches for multiphase flows.

perspectives on this intricate subject. We also made our attempt in Fig. 1.1, mostly inspired by [120] and the authors mentioned above.

The classification in Fig. 1.1 represents the most relevant methods suitable for simulating multiphase flows: in order to have a better understanding of SPH, we think that an excursus into the theoretical multiphase background would be beneficial. Multiphase flows can be simulated using various methods, including Lagrangian, Eulerian, or Arbitrary Lagrangian-Eulerian (ALE) techniques. In Lagrangian methods, the continuum is discretised with moving particles while in Eulerian approaches, a fixed grid is employed through which the flow passes. The ALE techniques fall in-between these two: they discretise the continuum with a movable grid that can change over time in a Lagrangian fashion. Needless to say that ALE methods tolerate well interface distortion.

Numerical simulations of multiphase flows may be challenging since the modelling involves coupling the continuum with the interface. The pioneer method for studying the interface was introduced by Harlow and Welch [84] and is known as the Marker-and-Cell (MAC) method. Initially developed for incompressible free-surface flows using finite-difference approximation, MAC utilises markers (passive tracers) to identify cells containing the fluid without actively participating in the calculations. One drawback of MAC was the occurrence of empty regions in the

domain, resulting in prolonged simulations; this issue was eventually addressed by incorporating dynamic grids.

Mesh-based models are classified based on the mathematical representation of phases (see 2021 publication by Soligo et al. [174]). The prevalent approach involves employing a single set of equations for the fluid domain, referred to as the one-fluid approach. Alternatively, body-fitted grid methods come into play when each phase is governed by its own set of equations, with coupling occurring at the interface; some useful connected references are Hirt et al. [86] and Ryskin and Leal [168, 167]. A third approach, as outlined in [174], is the sharp interface class (see the works by Glimm [72] and Fedkiw et al. [62]), which combines features from the previous two. In the sharp interface approach, each phase has its own set of Navier-Stokes equations, but they are implemented on a structurally simpler grid. While body-fitted methods offer high accuracy, they are more complex than the one-fluid methods.

Concerning the modelling of the interface, there are two main types of approaches: interface tracking and interface capturing. In the former, the interface is precisely identified using markers that move with it. In the latter, a phase indicator, often referred to as a color function, is employed, and a governing equation of the advection-diffusion type is formulated and solved.

Front-Tracking (FT) method was proposed by Unverdi and Tryggvason [190] in 1992 as a further development of the MAC method, and both are considered sharp interfaces. Similar to MAC, in Front-Tracking the interface is followed by a set of connected markers. Unfortunately, the FT is not well-suited for scenarios involving substantial deformations of the surface or changes in flow topology, such as break-ups or mergers.

The one-fluid class includes mainly four names: Volume of Fluid (VOF), Level Set (LS), Phase-field (PFM) and Constrained Interpolation Profile (CIP). These methods are categorised as interface capturing and each employs versions of a phase function.

The VOF is the oldest approach, introduced by Noh and Woodward [136]; initially employed the Simple Line Interface Calculation (SLIC), where the phase indicator took values of zero or one. In two dimensions, every cell is approximated by a straight line parallel to the y-axis for advection in x-axis and vice-versa. However, this early method lacked accuracy, resulting in a physically unrealistic broken interface. For this reason, a few years later, Youngs [200] suggested the interface reconstruction to improve the advection scheme. The straight-line segment in each cell was no longer aligned to the coordinate axis but determined by the normal to the interface itself. This method is called PLIC, piecewise linear interface calculation. Brackbill et al. [19] made a significant contribution to addressing the effects of surface tension with the introduction of the Continuum Surface Force (CSF). Before their work, surface tension force was typically ignored in implementing of the VOF. The addition of the

CSF term significantly improved results. Despite the complexity when of reconstructing the interface, especially in three dimensions, the VOF is widely implemented, even in commercial codes.

The Level Set method, conceived as an advancement of VOF a decade later, is associated with prominent researchers like Osher and Sethian [144], and Sussman et al. [180]. The first version of LS was based on the main idea by [144] that *"one wants to track the motion of a front whose speed depends on the local curvature"*, and eventually evolved into the modern LS. The phase indicator is a continuous function ranging from +1 to -1 and the isoline at the midpoint value identifies the interface. The smooth transition between phases provides higher numerical accuracy, but to ensure this advantage, the phase function must be reinitialised [180]. LS is robust and does not necessitate special treatments for coalescence and breakup phenomena. However, it suffers from issues of mass conservation when the characteristic size of two-phase flow structures becomes comparable. In such cases, small amounts of one phase mixed in the bulk of another phase lead to diffusion, resulting in under-resolved solutions.

The first work on the Phase-field (PF) method was presented by Cahn and Hilliard in 1958 [24] and later expanded upon in 1959 [23, 25]. In this study, the interface is defined by a finite thickness. While this transition layer between phases aligns with thermodynamics, it is considerably thicker than observed in real phenomena. The diffusive region is governed by the Cahn-Hilliard equation, which regulates the smearing effect, preventing it from becoming excessively large.

The Constrained Interpolation Profile method (CIP) [198], originally known as Cubic Interpolated Pseudo-particle method developed by Takewaki et al. [188], is a numerical solver that employs a Lagrangian invariant to transform the advection equation. It utilizes a cubic polynomial (or rational cubic polynomial) to solve the transformed equation.

Particle-based methods, although not as numerous as mesh-based ones, involve using a discrete set of fictitious particles to carry properties of real fluid particles. The most popular approach within the first category is the Lattice Boltzmann Method (LBM) [27], where particles are considered a coarse-grained group of fluid molecules. This method explores the mesoscopic behaviour of the fluid through the discrete Boltzmann equation. This equation defines the probability of a particle being at a specific position with a given velocity at a particular time, and how energy is distributed during particle collisions. LBM was first proposed by McNamara and Zanetti [117] (1988), and built upon previous mesoscopic methods like Lattice Gas Automation (LGA): instead of numbering the particles with integers, they tested the real numbers approach, resulting in a significant decrease in statistical noise. In summary, LBM is a rapid and easily parallelizable method, suitable not only for multiphase flows but also for flows with complex geometries such as porous media. Moreover, it is able to solve for heat transfer and chemical reactions in flows (see an

example in the 2013 paper by Grucelski and Pozorski [78]). However, it demands substantial memory resources during computation.

The second category includes the Moving-Particle Semi-Implicit (MPS) and the Smoothed Particle Hydrodynamics (SPH) methods. The MPS method was initially introduced in a 1996 article by Koshizuka and Oka [95] to model incompressible free-surface flows. While sharing many similarities with SPH, the key difference lies in how the pressure is implemented: given its incompressible nature, MPS employs the Poisson equation as the closure equation for pressure. The MPS approach has been further refined, and in 2016 Duan et al. [53] showed an extension of MPS to Multiphase MPS (MMPS). Previously, MPS encountered challenges associated with density and viscosity discontinuity at the interface. However, this issue was effectively addressed by enhancing the implementation of the pressure gradient in the momentum equation.

1.2 The Smoothed Particle Hydrodynamics approach

In 1977, a ground-breaking approach for astrophysics was simultaneously introduced by two distinct publications, one authored by Gingold and Monaghan [71] and the other by Lucy [110]. Within the context of astrophysics, where the significant deformation of stellar matter amplifies system complexity, a more in-depth exploration of fluid dynamics becomes imperative. Both research teams found that a Lagrangian approach offered a preferable solution. Additionally, the aim of this innovative particle-based method was to achieve reasonably accurate results using relatively low resolution. This led to the formulation of Smoothed Particle Hydrodynamics as a consequence of the Particle-in-Cell development.

Gingold and Monaghan [71] coined the term Smoothed Particle Hydrodynamics, drawing inspiration from smoothing kernel techniques. In SPH the particles are assigned with the value of fluid quantities, such as density for example, and they move in the computational domain according to the laws of dynamics.

In statistics, a kernel serves as a window function, being zero-valued beyond a chosen interval; normally symmetric, this function is usually at its maximum in the middle and it tapers off away from the centre. The term kernel has various meanings; for instance, it can be a weighting function to estimate the probability of density distribution, approximating the integral from the standard Monte Carlo (further details about Monte Carlo methods in the 2004 book by Hammersley and Handscomb [82]). As explained in the compendium by Monaghan [125], initially in [71, 110], SPH was envisioned as a Monte Carlo method due to its close ties with statistics: it was expected to behave as a probability function and to exhibit significant fluctuations, namely error, in the particle distribution.

However, this assumption was later contradicted when it was revealed that the error in SPH proved to be significantly smaller compared to Monte Carlo estimations. This discrepancy is attributed to the fact that SPH follows the dynamics.

In its early years, SPH was primarily implemented to solve problems on compressible fluids. Astrophysical applications were the most examined; the paper by Benz written in 1998 [10] is a comprehensive overview of the SPH capabilities during that time. In addition, the work by Evrard [59], explored the hydrodynamics of gas components in cosmology, while in the paper Nagasawa [133] presented the simulations on supernova explosion. Secondly, SPH proved to be well-suited for handling magnetohydrodynamics, as it is shown in the works by Phillips and Monaghan [149] and by Stellingwerf and Peterkin [178].

During the 1990s, SPH began expanding its influence into various other fields and gained global recognition. Libersky and Petschek [103] published in 1991 the first work on solid mechanics, testing a very young SPH for plastic deformation of an iron rod. In the next years, Benz and Asphaug [11] simulated the fracture of a basalt sphere, later Bonet and Kulasegaram [16] attempted a computation of metal forming.

For our purpose, one of the relevant publication is *Simulating free surface flow with SPH* by Monaghan [123]. Monaghan is one of the founding fathers of SPH, he implemented the approach to fluid dynamics simulating free surface phenomena, e.g. the benchmark of the dam break. The SPH of the nineties already embedded the basic formulas: the mass conservation, the equation of state for pressure that allows for weak compressibility and the momentum equation (with no surface tension). The author especially focused on the momentum, he added the artificial viscosity aware of the lack of inherent dissipation in SPH, and modified the advection term, naming it XSPH variant, to deal with noise in particle positions.

In 1995 Monaghan and Kocharyan [129] presented a dusty gas test case, paving the way for multiphase and sediment. Remarkable for two-phase flows are also the works by Colagrossi and Landrini [35] and Hu and Adams [87], where SPH was already very close to nowadays version. Moreover, in 2006 Zou and Dalrymple [204] proposed SPH solutions for sediment issues, while Chen et al. [26] attempted heat conduction. Finally, in 1997 the work by Morris et al. [132] showed the first implementation of incompressible fluids.

In the following years the SPH approach has been constantly improved and widened the number of study cases. The textbooks including the fundamentals on the methods are the one by Liu and Liu (2003) [107], the extensive monograph written by Violeau (2012) [192] and the work by Filho (2018) [65]. In addition, every now and then, review papers are published where the ongoing research is summarised, e.g. the compendium by Monaghan [125], the literature review in Crespo's manuscript [40],

the collection of coastal and ocean engineering applications presented by Gotoh and Khayyer [73] and the insight about accuracy by Lind et al. [106], and finally the SPH overview by Pozorski and Olejnik [151].

As mentioned in [106], Monaghan [126] was the first to propose a turbulence model in SPH in 2002, implementing a modified Lagrangian-Averaged Navier-Stokes. Differently, in 2014 Kajzer et al. [94] showed that the large eddy simulation (LES) method can be a viable option. This was confirmed a few years later by Di Mascio et al. [46], because the SPH smoothing can be viewed as LES Lagrangian filtering. However, LES application in SPH can result to be extremely computational expensive, and the subject is still matter of study.

Our team has consistently shown an interest in Lagrangian and particle approaches. In 1997 Minier and Pozorski [119] introduced a PDF model for turbulent flow, conceptualizing the fluid as a collection of particles. In 2002 Pozorski and Wawrenczuk [152] published the first article on SPH that involved simulations of lid-driven cavity flow and the Rayleigh-Taylor instability. A comprehensive understanding of our team's expertise can be gained from the doctoral thesis authored in 2013 by Szewc K. [182] followed by the 2019 dissertation by Olejnik M. [140]. Notably, our investigations have extended to micro-mixing issues in multi-phase problems (in 2016 by Szewc [183]), wetting phenomena, droplet-wall interactions (in 2019 by Olejnik [141]), and various multi-phase benchmarks.

1.3 Motivation and aim of this research

CFD for multiphase began with the MAC approach which has both Lagrangian and Eulerian characteristics, but over time, the predominant contribution has shifted towards grid-based methods. Mesh approaches perform well but exhibit drawbacks such as non-conservation of mass, face difficulties in handling large deformation (see the book by Manzari [113]) and in defining the interface. Generally, Lagrangian approaches are computationally expensive but also offer advantageous features, they are more intuitive, more straightforward to program, and robust, rarely suffer from mass conservation issues and eliminate the need to deal with mesh-related problems. Determining a superior approach between Eulerian and Lagrangian is challenging. Upon reviewing the FEM-SPH comparison in the work by Rakhsha et al. [156], it becomes evident that the choice should depend on the properties that effectively tackle the specific problem.

The comprehensive examination of SPH implementations in coastal and ocean engineering applications provided by [73] offers a thorough overview of the method's capabilities in handling fluid flows. Our team is particularly interested in SPH ability to simulate multiphase systems; for instance, in the 2016 publication by Olejnik et al. [142], the authors

have demonstrated the effectiveness of our in-house code in dealing with multiphase channel flows. Although the interface location in SPH does not require involve any specific treatment, the density and viscosity jump across the interface can require special attention, especially when both the density and the viscosity ratios are high.

Driven by the limited exploration of multiphase viscous flows in the current literature, our aim is to propose a developed model for interface description that adheres as much as possible to the laws of physics. Therefore, this manuscript evaluates the ongoing advancements in SPH and presents a novel formulation for the viscous force term in the momentum equation, assessing its potential applicability. The related goal is to demonstrate the capabilities of our code simulating the new model. Firstly, it is validated against relatively simple test cases, such as the steady-state Poiseuille flow in single- and two-phase scenarios, as well as the rising bubble. Then, it is tested against more complex benchmarks, including the sloshing phenomenon and the moon pool, a key feature in drill-ships and offshore platforms. We understand that our goal is ambitious, and SPH might not be fully ready for industry use. However, current research is progressing in this direction.

Theoretical background and governing equations

2

In this chapter, our aim is to explain the basics of SPH by introducing the interpolation method, the discretisation technique and the governing equations going into detail whenever a deeper insight into the matter is needed.

2.1 The interpolation method

2.1.1 Approximating a function in SPH

In SPH, the estimation of a scalar or vector field at a given position $\mathbf{r}$ involves an approximation process, with the initial reference being the following identity

$$A(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')d\mathbf{r}'. \tag{2.1}$$

The interpolation is expressed by the convolution of the field $A(\mathbf{r})$ and the Dirac delta function $\delta(\mathbf{r})$ integrated over the continuous medium Ω. The function $\delta(\mathbf{r})$ is a unit impulse; given that SPH discretises the continuum into a finite number of particles, a more effective interpolation is achieved through the utilization of a kernel function $W(\mathbf{r})$

$$\langle A(\mathbf{r})\rangle = \int_{\Omega} A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}' + \mathcal{O}(h^2). \tag{2.2}$$

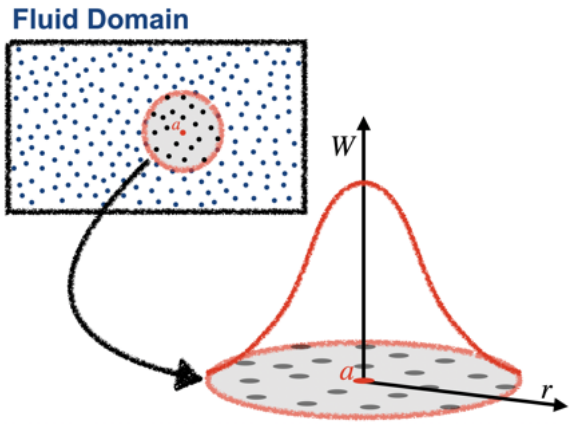


Figure 2.1: Schematic SPH representation of the fluid domain, bounded by the black rectangle: the fluid is discretised by a finite number of particles (blue points inside the rectangle). The fluid properties of a generic particle, denoted here with a, are determined by the correlation between the properties of the neighbouring particles and the kernel function W. The kernel function can be visualised with a bell-shaped pattern; the neighbouring particles are coloured grey.

2.1.2 Kernel interpolation

The function $W(\mathbf{r})$ acts as both a smoothing and weighting operator, with units corresponding to the inverse of the fluid volume. The kernel depends on the parameter h, referred to as the smoothing length. The kernel needs to meet several key conditions, as summarized in the 2003 work by Liu and Liu [108], outlined as follows.

1. It is normalised
$$\int_{\Omega} W(\mathbf{r}-\mathbf{r}', h)d\mathbf{r}' = 1. \tag{2.3}$$
2. It is symmetric with respect to the particle separation $\mathbf{r}-\mathbf{r}'$
$$W(\mathbf{r}-\mathbf{r}', h) = W(|\mathbf{r}-\mathbf{r}'|, h). \tag{2.4}$$
3. It must converge to $\delta(\mathbf{r})$ when $h \to 0$.
4. It is a positive weighting function.
5. It monotonically and symmetrically decreases with the relative distance $\mathbf{r}-\mathbf{r}'$, typically following a bell-shaped pattern.
6. It is "sufficiently smooth", meaning the central peak is large enough so as not to be too sensitive to small changes of particle positions and has a better accuracy.
7. It is at least twice differentiable.
8. Preferably, it is computationally inexpensive.
9. It has a compact support that can be visualised as the radius of its bell-shaped pattern. The compact support is a multiple of the smoothing length, equal to $\kappa \cdot h$, where $\kappa = \mathcal{O}(1)$, and holds the following conditions:
$$W(\mathbf{r}-\mathbf{r}', h) = 0 \qquad \text{for} \qquad |\mathbf{r}-\mathbf{r}'| > \kappa h \tag{2.5}$$
$$W(\mathbf{r}-\mathbf{r}', h) \geq 0 \qquad \text{for} \qquad |\mathbf{r}-\mathbf{r}'| \leq \kappa h. \tag{2.6}$$
10. It should have the form
$$W(\mathbf{r}-\mathbf{r}', h) = \alpha_d \mathrm{f}(q) \tag{2.7}$$
where α_d is the dimension-dependent normalisation constant for a given kernel and $q = |\mathbf{r}-\mathbf{r}'|/h$.

Although there is no function that perfectly aligns with this list of criteria, various promising applications have been tested in the literature. Overall, there are three main types of smoothing kernel (see Antuono [3] overview presented in 2019): Gaussian-type, B-spline kernels (the first suitable because differentiable two times is the cubic spline), and polynomial type (e.g., Wendland kernels).

The Gaussian kernel appears to be the most suitable choice as it meets nearly all the requirements except the finite compact support. This condition is rather important from the simulation perspective since employing the Gaussian interpolation involves the interaction of all the

particles in the domain, causing a significant increase in computational time. A preferable alternative would be a Gaussian-like function truncated at a finite radius, which is proportional to the scaling parameter h. The compact support establishes the region of influence of the kernel, and despite making it more susceptible to local particle distribution, it considerably decreases the computational cost.

Neither the B-splines nor the Wendland functions were designed on purpose for the SPH needs. The piecewise function family consisting of the Basis Functions of splines curve (from which the abbreviation B-spline, by Schoenberg [170]) were primarily intended for interpolating polynomials on a regular one-dimensional grid rather than accommodating particle distributions in two or three dimensions. Whereas the Wendland functions are Radial Basis Functions with relatively low order polynomials, have compact support, and were developed (Wendland, 1995 [194]) to interpolate multi-dimensional data or solve partial differential equations thanks to their meshless nature. Nevertheless, both B-splines and Wendland functions have found widespread application.

The smoothing length h is a fundamental parameter; it is comparable to a spatial filter, defining how many particles are involved in the kernel interpolation. In incompressible or weakly-compressible fluid dynamics, it typically remains constant, while for systems where density is characterised by high variability it must be defined differently. For example, as described in the 2012 publication by Price, in astrophysics, the smoothing length is indirectly related to the local particle distribution. Therefore, if d the spatial dimension, h is correlated with the local number density, referred as θ

$$h(\mathbf{r}) \propto \langle \theta(\mathbf{r})^{-1/d} \rangle \tag{2.8}$$

2.1.3 Instabilities in SPH

Two main types of numerical instabilities have been identified:

- The tensile instability that occurs when the SPH pressure yields a negative value. This leads the particles to attract rather than repel each other and to form clusters with adverse and undesired consequences on the simulation result.
- The pairing instability occurs when particles form a close pair, reducing the near-neighbour force.

The tensile instability was first consistently analysed in 1995 by Swegle et al. [181]; in 2000, Monaghan [127] proposed a solution by adding an artificial diffusive term in the momentum equation called artificial viscosity. Other arrangements have eventually been proposed, e.g., for truly incompressible SPH and structural mechanics. In our formulation, we do not use the artificial viscosity (details in [182]).

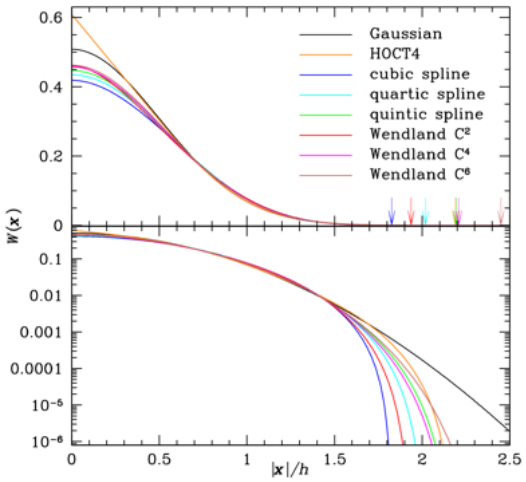
Commonly, it was believed that the pairing instability resulted from a reduction of the repulsive force between neighbouring particles, which arises from smoothing kernels with an inflection point. However, in 2012 Dehnen and Aly [45] refuted this idea: through a systematic analysis of the pairing instability problem, the authors demonstrated that there exist stable kernels with an inflection point as well. In SPH, the particles are not randomly placed but they rather follow a semi-regular glass-like distribution. This characterisation does not prevent from having some degree of particle disorder, which inherently induces a random error in the pressure approximation that slows down the convergence. To the author's knowledge, the most reasonable measure to diminish the error is to increase the number of particles involved in the interpolation, denoted as the neighbouring particles N_{neigh}, in two ways: either by changing the kernel and choosing one with larger support, or by stretching the smoothing length h. However, a large N_{neigh} can cause kernel instability and lead to the issue of pairing, in particular in B-spline kernels, as confirmed also in the work by Szewc et al. [184]. For this reason, the question in [45] was: why do some kernels retain the stability when N_{neigh} increases while the splines do not? In the paper, three different types of kernels are compared: the B-splines (cubic, quartic and quintic) and the Wendland functions (of orders C^2, C^4 and C^6),that have the inflection point versus the triangular kernel, which lacks it. A viable option for a triangular kernel was investigated by Read et al. [160]. Referred to as HOCT4, this function can be visualised as a triangular-shape bell and demonstrates stability even with large values of N_{neigh}. On the other hand, examples of B-spline and Wendland functions are collected in Tab. 2.1 along with their specifications, where B^{-d} represents the kernel support radius, C' the normalisation constant and $w(q)$ the kernel function.

$$W(q,h) = B^{-d} C' w(q) \tag{2.9}$$

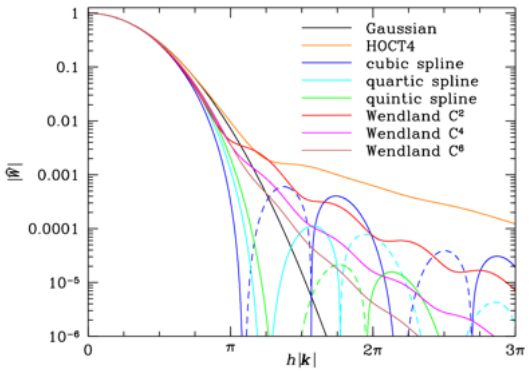
Table 2.1: B-spline and Wendland kernel functions along with their specification, table from [45]. In this table $d = 1, 2, 3$ denotes the spatial dimension, $(\cdot)_+ \equiv \max\{0, \cdot\}$, C' the normalisation constant and $w(q)$ the kernel function. The formulations of Wendland are identical for $d = 2$ and $d = 3$, while it differs in case of $d = 1$.

kernel name	$w(q)$	C'		
cubic spline	$(1-q)_+^3 - 4\left(\frac{1}{2}-q\right)_+^3$	$\frac{8}{3}$	$\frac{80}{7\pi}$	$\frac{16}{\pi}$
quartic spline	$(1-q)_+^4 - 5\left(\frac{3}{5}-q\right)_+^4 + 10\left(\frac{1}{5}-q\right)_+^4$	$\frac{5^5}{768}$	$\frac{5^6 3}{2398\pi}$	$\frac{5^6}{512\pi}$
quintic spline	$(1-q)_+^5 - 6\left(\frac{2}{3}-q\right)_+^5 + 15\left(\frac{1}{3}-q\right)_+^5$	$\frac{3^5}{40}$	$\frac{3^7 7}{478\pi}$	$\frac{3^7}{40\pi}$
Wendland $C^2, d=1$	$(1-q)_+^3(1+3q)$	$\frac{5}{4}$	-	-
Wendland $C^4, d=1$	$(1-q)_+^5\left(1+5q+8q^2\right)$	$\frac{3}{2}$	-	-
Wendland $C^6, d=1$	$(1-q)_+^7\left(1+7q+19q^2+21q^3\right)$	$\frac{55}{32}$	-	-
Wendland $C^2, d=2,3$	$(1-q)_+^4(1+4q)$	-	$\frac{7}{\pi}$	$\frac{21}{2\pi}$
Wendland $C^4, d=2,3$	$(1-q)_+^6\left(1+6q+\frac{35}{3}q^2\right)$	-	$\frac{9}{\pi}$	$\frac{495}{32\pi}$
Wendland $C^6, d=2,3$	$(1-q)_+^8\left(1+8q+25q^2+32q^3\right)$	-	$\frac{78}{7\pi}$	$\frac{1365}{64\pi}$

The kernel functions are compared in Fig. 2.2a, where they are represented in linear and logarithmic scale. Clearly, the spline functions exhibit lower peaks; the Wendland functions closely resemble the Gaussian distribution, and finally, the HOCT4 kernel has linear behaviour. The Fourier transform of each kernel is estimated as a function of the smoothing length in the frequency domain (where **k** is a wave vector), and they are plotted in Fig. 2.2b. Among the functions, B-spline types are mostly susceptible to pairing instability. This characteristic is evident in the plot, where negative Fourier transforms, depicted by dashed curves, indicate oscillations around zero. Conversely, the Wendland kernels show more stability when varying N_{neigh}: they are Gaussian-like functions with compact support, they have a significant smoothing effect and are originally designed for multi-dimensional scatter data. Therefore, [45] has demonstrated that the Wendland functions are a convenient option.



(a) On the top: linear plot of functions in Tab. 2.1, where $\mathbf{x} = \mathbf{r} - \mathbf{r}'$ is adimensionalised relative position of particles and arrows indicate $|\mathbf{x}| = B$. On the bottom: logarithmic plot.



(b) Fourier Transform $|\hat{W}|$ of kernel functions in 2.1, negative values are plotted with broken curves.

Figure 2.2: Comparison of kernel functions by [45]. (a) Linear (top) and logarithm (bottom) scale representation of kernel types, considering as reference the Gaussian function. (b) Representation of Fourier transform of kernel functions, negative values of spline functions denote oscillations, therefore presence of pairing instability.

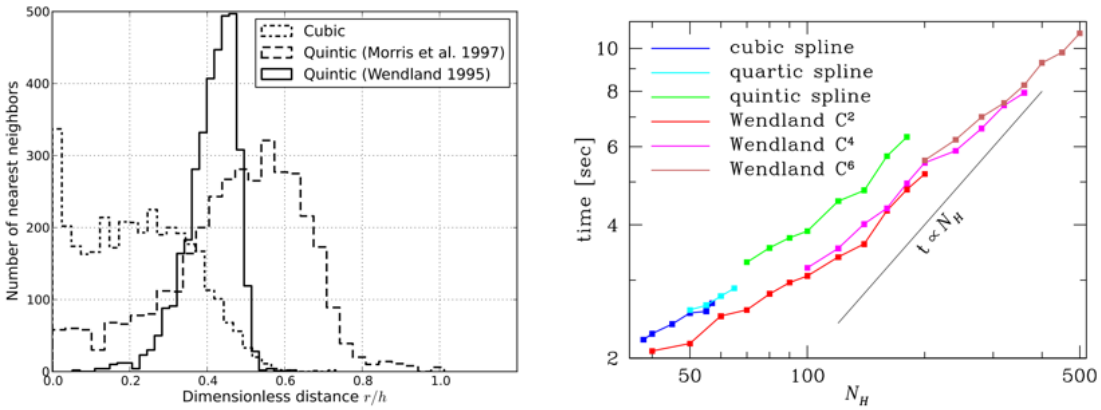
2.1.4 The kernel in our in-house code

We opted for the quintic Wendland kernel C^2, employed in the two-dimensional form denoted with $d = 2$; the formulation is

$$W(\mathbf{r}, h) = B^{-d} C' \begin{cases} \left(1 - \frac{q}{2}\right)^4 (2q + 1), & \text{for } q < 2 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

in which $q = |\mathbf{r} - \mathbf{r}'|/h = \Delta r/h$, the support has $B = 2h$ radius and $C' = 7/\pi$ is the normalisation constant. In the case of three dimensions, meaning with $d = 3$, the normalisation constant is equal to $C' = 9/\pi$. In [184], conducting benchmark tests verified that there is no pairing

occurring for this specific kernel. Moreover, Fig.2.3 provides further information about the correlation between the kernel function and the number of neighbouring particles N_{neigh}. Based on the statistical validation done by [184] and represented in Fig. 2.3a, the typical distance between adjacent particles for the quintic Wendland kernel is approximately $\Delta r = h/2$, which is also the initial condition in our in-house code. Looking at Fig. 2.3b the time is dependent on N_{neigh}, indicating that Wendland functions are more suitable for interpolating large numbers of particles.



(a) Histogram of the distance between the nearest pairs of particles, more frequent distance is $\Delta r = h/2$. Taken from [184].

(b) Wall-clock timings for a single time step produced with the kernels in Tab.2.1, where$N_h = N_{neigh}$ is the number of neighbour particles. Plot from [45].

Figure 2.3: Comparison of kernel functions. Respectively: (a) distance between the nearest pairs of particles; (b) timing dependence on number of neighbour particles.

Even though an exact value for N_{neigh} has not been universally defined, the number of particles cannot be too small due to resolution nor too high due to computational time. In conclusion, using the Wendland kernel with $q = 2$ and h constant, we usually have for simulations in $d = 2$ a minimum $N_{neigh} \approx 45$, which can increase to a few more particles based on the local distribution.

2.1.5 First order derivative

1: Following equations refer to content from page 307 in [192].

The first order derivative of the arbitrary field $A(\mathbf{r})$ of a continuous medium is[1]

$$\langle \nabla A(\mathbf{r}) \rangle \approx \int_{\Omega_r} \frac{\partial A(\mathbf{r}')}{\partial \mathbf{r}'} W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' \tag{2.11}$$

where $\Omega_\mathbf{r}$ is the portion of the whole domain Ω defined by the compact support and with the center at the point $\mathbf{r}$. Proceeding to the derivation, the above equation can be written as follows:

$$\langle \nabla A(\mathbf{r}) \rangle \approx \int_{\Omega_r} \frac{\partial}{\partial \mathbf{r}'} [A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h)] d\mathbf{r}' - \int_{\Omega_r} A(\mathbf{r}') \frac{\partial W(\mathbf{r} - \mathbf{r}', h)}{\partial \mathbf{r}'} d\mathbf{r}'. \tag{2.12}$$

With the assumption of the following identity

$$\frac{\partial W(\mathbf{r}-\mathbf{r}',h)}{\partial \mathbf{r}'} = -\frac{\partial W(\mathbf{r}-\mathbf{r}',h)}{\partial \mathbf{r}} \tag{2.13}$$

the equation then reads

$$\langle \nabla A(\mathbf{r}) \rangle \approx \oint_{\partial\Omega_\mathbf{r}} A(\mathbf{r}') W(\mathbf{r}-\mathbf{r}',h)\mathbf{n}(\mathbf{r}')d\Gamma + \int_{\Omega_\mathbf{r}} A(\mathbf{r}')\frac{\partial W(\mathbf{r}-\mathbf{r}',h)}{\partial \mathbf{r}}d\mathbf{r}' \tag{2.14}$$

where $\mathbf{n}(\mathbf{r}')$ is the outward unit vector normal to the boundary of $\Omega_\mathbf{r}$, while Γ represents the boundary of the domain. The contour integral in the RHS results from the intersection of the kernel compact support with the boundary; if the point $\mathbf{r}$ is far enough from $\partial\Omega$, the intersection is null as is the integral. Hence, the final formulation is

$$\langle \nabla A(\mathbf{r}) \rangle \approx \int_{\Omega_\mathbf{r}} A(\mathbf{r}')\nabla W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}'. \tag{2.15}$$

The divergence operator of an arbitrary vector field $\mathbf{A}(\mathbf{r})$ is very similar, and is written as:

$$\langle \nabla \cdot \mathbf{A}(\mathbf{r}) \rangle \approx \int_{\Omega_\mathbf{r}} \mathbf{A}(\mathbf{r}') \cdot \nabla W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}'. \tag{2.16}$$

2.1.6 Discretisation technique

The particles that discretise the fluid seem to move like material points, but each of them is a macroscopic entity with fixed mass m (or volume V). Indeed, in Colagrossi et al. [37], the volume of an arbitrary particle a can be expressed as $V_a = 1/\sum_b W_{ab}$ where the neighbouring particles are denoted with the subscript b. If the particle has a sphere-like shape, its size is $\delta r_a = \sqrt[3]{V_a}$; whenever the size of the particle decreases towards zero $\delta r \to 0$ the number of particles in the domain must increase to infinite $N \to \infty$ transitioning from discrete to continuous form. Hence, the physical volume of the entire domain is only approximated by summing the particle volumes $\Omega \approx \sum_b V_a$.

The discrete equation of the SPH interpolation of the field $A(\mathbf{r}_a)$ is If a is an arbitrary particle in the domain, and b is a neighbour one,

$$\langle A(\mathbf{r}) \rangle_a = \sum_b A(\mathbf{r}_b) W(\mathbf{r}_a - \mathbf{r}_b, h)\frac{m_b}{\rho_b}. \tag{2.17}$$

while the derivative can be written as

$$\langle \nabla A(\mathbf{r}) \rangle_a = \sum_b A(\mathbf{r}_b) \nabla W(\mathbf{r}_a - \mathbf{r}_b, h)\frac{m_b}{\rho_b}. \tag{2.18}$$

Despite the formal correctness of the latter, it has been proven that this formulation is not the most convenient in SPH.

A generalised version of the gradient operator can be derived using the following identity, from which we can get the two variants, antisymmetric and symmetric expression, respectively written as

$$\nabla A = \nabla\left(\rho^{\alpha}\frac{A}{\rho^{\alpha}}\right) \overset{antisym}{=} \rho^{\alpha}\nabla\left(\frac{A}{\rho^{\alpha}}\right) + \frac{A}{\rho^{\alpha}}\nabla(\rho^{\alpha}) \overset{sym}{=} \frac{1}{\rho^{\alpha}}\nabla(\rho^{\alpha}A) - \frac{A}{\rho^{\alpha}}\nabla(\rho^{\alpha}). \tag{2.19}$$

The antisymmetric discrete formulation reads

$$\langle\nabla A(\mathbf{r})\rangle_a = \sum_b \frac{\rho_a^{2\alpha}A_b + \rho_b^{2\alpha}A_a}{\rho_a^{\alpha}\rho_b^{\alpha}}\nabla W_{ab}\frac{m_b}{\rho_b} \tag{2.20}$$

and the symmetric one is

$$\langle\nabla A(\mathbf{r})\rangle_a = -\frac{1}{\rho_a^{2\alpha}}\sum_b(\rho_a\rho_b)^{\alpha}A_{ab}\nabla W_{ab}\frac{m_b}{\rho_b} \tag{2.21}$$

where $W_{ab} = W(\mathbf{r}_a - \mathbf{r}_b, h)$, $A_a = A(\mathbf{r}_a)$ and $A_{ab} = A(\mathbf{r}_a) - A(\mathbf{r}_b)$, α is an integer value, e.g. zero or one. We will explain in section 2.2.6 why the antisymmetric formulations should be preferred.

The formulae for the divergence operators resemble what has been shown so far for the gradient, but between the field A and the kernel it is implemented a scalar product instead of a dyadic product. Thus, the antisymmetric form of the divergence reads

$$\langle\nabla\cdot\mathbf{A}(\mathbf{r})\rangle_a = \sum_b \frac{\rho_a^{2\alpha}\mathbf{A}_b + \rho_b^{2\alpha}\mathbf{A}_a}{\rho_a^{\alpha}\rho_b^{\alpha}}\nabla\cdot W_{ab}\frac{m_b}{\rho_b} \tag{2.22}$$

while the symmetric is

$$\langle\nabla\cdot\mathbf{A}(\mathbf{r})\rangle_a = -\frac{1}{\rho_a^{2\alpha}}\sum_b(\rho_a\rho_b)^{\alpha}\mathbf{A}_{ab}\nabla\cdot W_{ab}\frac{m_b}{\rho_b}. \tag{2.23}$$

2.1.7 Second order derivative discrete formulation

The second order derivative, or Laplacian operator, in the discrete form could be

$$\langle\nabla^2 A(\mathbf{r})\rangle_a = \sum_b A_b\nabla^2 W_{ab}\frac{m_b}{\rho_b} \tag{2.24}$$

but in this variant, the kernel derivative was revealed to be sensitive to particle disorder, especially for low-order kernels. In addition, in [125], it is proved that a few other drawbacks are related: the exact second derivative can not assure the antisymmetry formulation, that is, the discrete conservation of $A(\mathbf{r})$; based on the type of the kernel, this operator can become negative, which leads to a wrong physical interpretation for some quantities (e.g. temperature in the energy equation). There are a couple of other theoretically simple alternatives, each involving a double summation formula. However, it is evident that incorporating

an additional summation in the calculation causes increased simulation time.

In SPH is preferred a different version, precisely the one that employs the following approximation of the arbitrary vector field $\mathbf{A}$. This approximation is derived from the Taylor series expansion of the field around the point $\mathbf{r}_a$

$$\mathbf{A}_b = \mathbf{A}_a - (\nabla \mathbf{A})_a \cdot \mathbf{r}_{ab} + \mathcal{O}(r_{ab}^2) \tag{2.25}$$

where $\mathbf{r}_{ab} = \mathbf{r}_a - \mathbf{r}_b$ is the relative distance between two particles. The divergence of the field can also be written as follows

$$(\nabla \mathbf{A})_a \cdot \mathbf{e}_{ab} \approx \frac{\mathbf{A}_{ab}}{r_{ab}}. \tag{2.26}$$

where $\mathbf{e}_{ab}$ is the unit vector of $\mathbf{r}_{ab}$. Assuming the following identity holds true

$$(\nabla \mathbf{A})_b \cdot \mathbf{e}_{ab} = -(\nabla \mathbf{A})_b \cdot \mathbf{e}_{ba} \approx -\frac{\mathbf{A}_{ba}}{r_{ba}} = \frac{\mathbf{A}_{ab}}{r_{ab}} \tag{2.27}$$

we can write a generic version of the second order operator

$$\langle \nabla (J_{\mathbf{A}} \nabla \mathbf{A}) \rangle_a \approx \sum_b \frac{\rho_b^{2\alpha} J_{\mathbf{A},a} + \rho_a^{2\alpha} J_{\mathbf{A},b}}{(\rho_a \rho_b)^{\alpha}} \frac{\mathbf{A}_{ab}}{r_{ab}} \frac{m_b}{\rho_b} \nabla W_{ab} \tag{2.28}$$

where $J_{\mathbf{A}}$ is the diffusion coefficient and α is an integer number (for instance equal to zero or one). The same approach can be applied to a scalar field too.

However, an alternative approximation has taken hold over the years, inspired by the 2005 work by Monaghan [125] and the 2003 research by the authors Español and Revenga [58]. This proposal is obtained by the Taylor expansion of $A(\mathbf{r})$ around $\mathbf{r}$ (explained in the book by Violeau [192]). When considering weak compressibility and accounting for the geometric dimension d, the second-order derivative is written as:

$$\langle \nabla (J_{\mathbf{A}} (\nabla \mathbf{A})^T) \rangle_a \approx \sum_b \frac{\bar{J}_{A,ab}}{r_{ab}} [(d+2)(\mathbf{A}_{ab} \cdot \mathbf{e}_{ab}) \mathbf{e}_{ab} - \mathbf{A}_{ab}] W'_{ab} \tag{2.29}$$

taking $\nabla W_{ab} = W'_{ab} \mathbf{e}_{ab} = \partial W_{ab} / \partial r_{ab} \mathbf{e}_{ab}$, the versor $\mathbf{e}_{ab} = \mathbf{r}_{ab} / r_{ab}$ and $\bar{J}_{A,ab}$ the mean value of the diffusion coefficient.

2.1.8 Accuracy in SPH: error definition

In mesh-based methods the discretisation technique defines the so-called *discretization error* which simply is the difference between the numerical solution to the discretised equations and the theoretical solution to the partial differential (or integral) equations. Looking at Eq. (2.2) in the continuous form and its discrete form Eq.(2.17), we convene that in SPH the error is due to the smoothing procedure, that in practice depends

on particles number and distribution, and on the smoothing kernel. The estimation of the error comes from expanding the function $A(\mathbf{r})$ in a Taylor series about $\mathbf{r}$. In general, in SPH the error is defined as a function of the square of the smoothing length. We clearly see this in the work by Price [153] published in 2012, where the Taylor expansion of $A(\mathbf{r})$ reads

$$\langle A(\mathbf{r})\rangle = \int \left[A(\mathbf{r}) + (\mathbf{r}' - \mathbf{r})^\alpha \frac{\partial A}{\partial \mathbf{r}^\alpha} + \frac{1}{2}(\mathbf{r}' - \mathbf{r})^\alpha (\mathbf{r}' - \mathbf{r})^\beta \frac{\partial^2 A}{\partial \mathbf{r}^\alpha \partial \mathbf{r}^\beta} + \mathcal{O}(h^3)\right] W(\mathbf{r} - \mathbf{r}')d\mathbf{r}' \tag{2.30}$$

and if it holds the kernel properties of symmetry and normalisation, we have

$$\langle A(\mathbf{r})\rangle = A(\mathbf{r}) + \frac{1}{2}\frac{\partial^2 A}{\partial \mathbf{r}^\alpha \partial \mathbf{r}^\beta} \int (\mathbf{r}' - \mathbf{r})^\alpha (\mathbf{r}' - \mathbf{r})^\beta W(|\mathbf{r}' - \mathbf{r}|)d\mathbf{r}' + \mathcal{O}(h^4).$$

Hence, if the kernel is second order differentiable the interpolation is second order accurate [$\mathcal{O}(h^2)$]. In the discrete formulation the Taylor series of A_a is

$$\langle A_a\rangle = \sum_b m_b \frac{A_b}{\rho_b} W_{ab} = A_a \sum_b \frac{m_b}{\rho_b} W_{ab} + \nabla A_a \cdot \sum_b \frac{m_b}{\rho_b}(\mathbf{r}_a - \mathbf{r}_b) W_{ab} + \mathcal{O}(h^2) \tag{2.31}$$

which is truly second order if the following conditions are verified

$$\sum_b \frac{m_b}{\rho_b} W_{ab} \approx 1 \tag{2.32}$$

$$\sum_b \frac{m_b}{\rho_b}(\mathbf{r}_b - \mathbf{r}_a) W_{ab} \approx 0. \tag{2.33}$$

In practice, the circumstance of which the approximation accuracy respects the second order strongly depends on the uniform distribution of the particles which, as already mentioned, is preserved when instabilities are absent. Similarly, the Taylor series of the continuous interpolation gradient is [153]

$$\nabla A(\mathbf{r}) = \nabla A(\mathbf{r}) + \frac{1}{2}\frac{\partial^2 A}{\partial \mathbf{r}^\alpha \partial \mathbf{r}^\beta} \int (\mathbf{r}' - \mathbf{r})^\alpha (\mathbf{r}' - \mathbf{r})^\beta \nabla W(|\mathbf{r} - \mathbf{r}'|)d\mathbf{r}' + \mathcal{O}(h^3) \tag{2.34}$$

while the discrete formulation, differentiated α times, reads

$$\langle \nabla A_a\rangle = A_a \sum_b \frac{m_b}{\rho_b} \nabla_a W_{ab} + \frac{\partial A_a}{\partial \mathbf{r}^\alpha} \sum_b \frac{m_b}{\rho_b}(\mathbf{r}_b - \mathbf{r}_a)^\alpha \nabla W_{ab} + \mathcal{O}(h^2) \tag{2.35}$$

being true if the conditions below hold

$$\sum_b \frac{m_b}{\rho_b} \nabla W_{ab} \approx \mathbf{0} \tag{2.36}$$

$$\sum_b \frac{m_b}{\rho_b} (\mathbf{r}_b - \mathbf{r}_a)^\alpha \nabla^\beta W_{ab} \approx \delta^{\alpha\beta}. \tag{2.37}$$

where δ is the Kronecker symbol. A comprehensive insight on how the error is exactly calculated is also included in the book [192].

During the seminar in Delft in 2019, a summary of the convergence requirements was provided in a presentation titled *The Mathematics of SPH* delivered by Antuono [3]. Therefore, when considering a particle belonging to the fluid domain, the difference between exact and discrete formulation depends on the inter-particle distance and the particle distribution, as shown in Fig. 2.4.

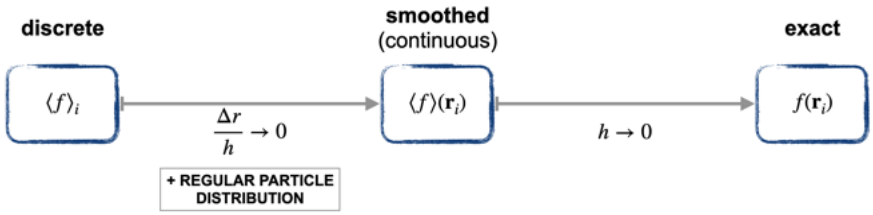


Figure 2.4: Scheme from [3]. The error decreases from left to right until reaching the exact convergence. This condition holds true if $r_i \in \Omega \subset D$ (i.e. inside the fluid domain).

We tested the rate of convergence of the SPH interpolation in one dimension (see Fig.2.5); the kernel used is the quintic Wendland, and the corresponding smoothing length is stretched from the reference value twice the particle distance (here indicated with δr) to five times, which translates in a higher number of particles under the kernel bell. We see that the number of particles is not trivial; in fact, the convergence gets faster with the increment of particles.

Recently, the investigation of the error has made progress, and we consider the work by Violeau and Fonty [193] inspiring. The authors prove the error to be proportional to the kernel standard deviation σ, previously defined in Dehnen and Aly [45] as

$$\sigma_{\mathbf{r}}^2 = \frac{1}{d} \int (\mathbf{r} - \mathbf{r}')^2 W(\mathbf{x} - \mathbf{x}') d^d \mathbf{x}'. \tag{2.38}$$

The element $\sigma_{\mathbf{r}}$ is, in fact, the correct length to measure the smoothing distance in SPH, while the smoothing length is established according to the type of kernel, and it makes the estimate of the interpolating error almost kernel-independent. In their study, any SPH operator is turned in a function of the complex domain through the Laplace transform, which conveniently switches convolutions with multiplications. Finally, the differential operators are functions of the kernel standard deviation, cutting out the kernel.

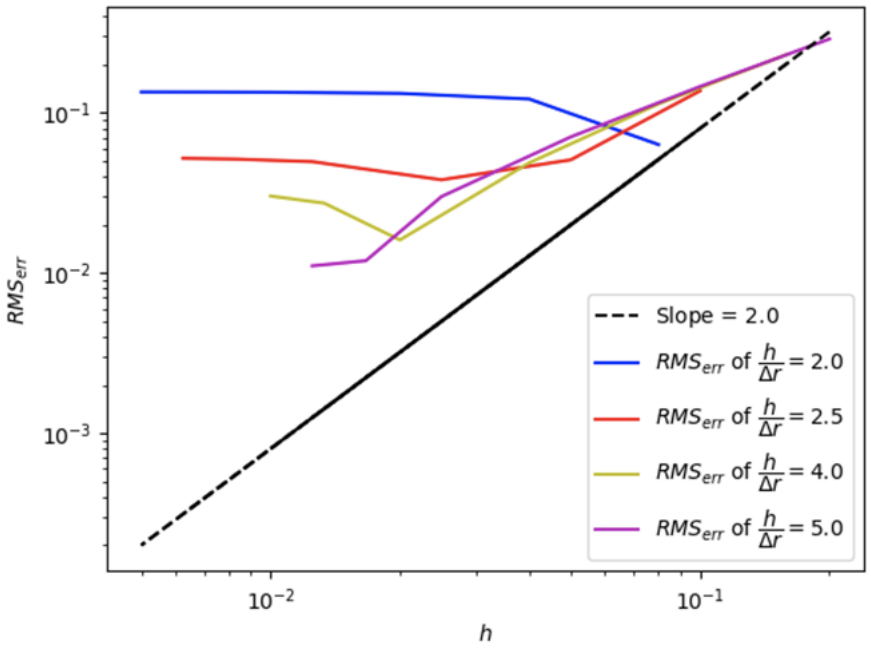


Figure 2.5: Rate of convergence of one-dimensional interpolant summation in SPH, logarithmic scale: black line represents the order of accuracy $\mathcal{O}(h^2)$ while coloured lines account for kernels with increasing support. Respectively, from $h/\delta r = 2$ to $h/\delta r = 5$ the number of particles under the kernel bell counts 25, 50, 100, 200, 300 and 400. Visibly, the increment of particles under the kernel allows the function to converge.

2.2 Foundation of the physics in SPH

2.2.1 SPH core: the density definition

In SPH, the continuum can be represented by particles with assigned fixed mass; the way the particles affect each other is determined by the kernel function that interpolates smoothly the particle quantities based on the scaling parameter h [153].

In contrast to grid-based and other Lagrangian particle methods where the discretisation procedure aligns with volume, in SPH, it relies on density treatment. Notably, the density does not depend on the absolute position of the particles and on their time evolution but rather on the relative distances between particle pairs, as reflected in the discrete equation

$$\langle \rho(\mathbf{r}) \rangle = \sum_b m_b W(\mathbf{r}_a - \mathbf{r}_b, h). \tag{2.39}$$

As a result, the conservation of total mass is assured based on the density definition. The computation of density not only represents the Lagrangian behaviour of particles but also indicates how physical laws are handled within SPH.

The most significant properties of the discretisation technique are outlined below (from [154]):

1. the exact solution to the continuity equation;
2. advection done perfectly;
3. zero intrinsic dissipation;
4. exact and simultaneous conservation of mass, momentum, angular momentum, and energy;

5. guaranteed minimum energy state for particles;
6. resolution that follows the mass.

In the following, we will initially explain how the Lagrangian formalism is adapted to SPH, practically replaced by the Hamiltonian, and finally what are the physical consequences of this choice.

2.2.2 From the Lagrangian to Hamiltonian formalism

A way to describe the dynamics of a system is through the equation of motion. We can construct this equation using Newton's second law, which is only suitable for systems with inertial coordinates [2]. However, when dealing with systems composed of particles, such as in SPH, the Lagrangian formalism is a more appropriate selection [153]. In classical mechanics, the Lagrangian formulation

2: Theoretical background about Lagrangian and Hamiltonian formulations can be found in the Hamill's textbook [80].

$$\mathscr{L} = U_{kin} - U_{pot} \tag{2.40}$$

is a subtraction of the kinetic energy U_{kin} and potential energy U_{pot} of a system. If $\mathbf{v} = d\mathbf{r}/dt$ is the velocity of an arbitrary particle a, the Lagrangian in thermodynamics can be written as

$$\mathscr{L} = \sum_a m_a \left[\frac{1}{2}v_a^2 - U_a(\rho_a, S_a)\right] \tag{2.41}$$

where v_b is the velocity of particle b, U_b is the thermal energy specified as a function of the density ρ_b and entropy S_b.

In Cartesian coordinates, the Lagrangian relies on position (through the energy U), velocity $\mathbf{v}$, and is generally time-variable. A further way to express the Lagrangian is using the generalised coordinates: q_i represents the generalised position of the i-th particle, while $\dot{q}_i$ denotes the generalized velocity. Thus, the Lagrangian of a particle results in the function $\mathscr{L} = \mathscr{L}(q_i, \dot{q}_i, t)$. The Lagrange equations are derived from Hamilton's principle, which refers to the least action principle. Essentially, Hamilton's principle states that the path followed by a system minimises the action, denoted as $\mathscr{I}$. Consequently, we can express this as:

$$\mathscr{I} = \int \mathscr{L}\, dt \qquad \text{with} \qquad \mathscr{L} = \int [\rho v^2 - \rho U(\rho, S)]\, dV \tag{2.42}$$

to minimise the action, the variation $\partial\mathscr{I}$ is zero

$$\partial\mathscr{I} = \partial \int \mathscr{L}\, dt = \int \left(\frac{\partial\mathscr{L}}{\partial\mathbf{v}} \cdot \delta\mathbf{v} + \frac{\partial\mathscr{L}}{\partial\mathbf{r}} \cdot \delta\mathbf{r}\right) dt = 0. \tag{2.43}$$

where δ represent a small variation. Applying the velocity identity $\delta\mathbf{v} = d(\delta\mathbf{r})/dt$ and the material derivative $D\cdot/Dt = \partial\cdot/\partial t + \mathbf{v}\cdot\nabla$ we

can re-write Eq. 2.43 as

$$\partial \mathcal{I} = \int \left\{ \left[-\frac{D}{Dt}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{v}}\right) + \left(\frac{\partial \mathcal{L}}{\partial \mathbf{r}}\right) \right] \cdot \partial \mathbf{r} \right\} dt + \left[\frac{\partial \mathcal{L}}{\partial \mathbf{v}} \cdot \partial \mathbf{r} \right]_{t_0}^{t} = 0. \quad (2.44)$$

If the variation in the second term on the RHS vanishes at the initial time t_0 and at the final time t, we can ultimately formulate the Euler-Lagrangian equation with respect to a particle

$$\frac{D}{Dt}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{v}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{r}} = 0. \quad (2.45)$$

This equation remains valid under three conditions: the time derivative is computed exactly; the Lagrangian density and thermal energy are computed exactly; the entropy is conserved, that translates as the lack of dissipation. In the presence of discontinuity, special measures are required to smooth them out to the scale of resolution h, effectively eliminating them as discontinuities in the formulation. The application of the Legendre transformation to the Lagrangian gives $\mathcal{H}$, the Hamiltonian

$$\mathcal{H} = \sum \mathbf{v} \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{v}} - \mathcal{L}. \quad (2.46)$$

Using the generalised coordinates rather than the Cartesian coordinates, the term $\partial \mathcal{L}/\partial \mathbf{v}$ becomes $\partial \mathcal{L}/\partial \dot{q}$ and the Hamiltonian dependencies are $\mathcal{H} = \mathcal{H}(q, \partial \mathcal{L}/\partial \dot{q}, t)$.

3: Canonical equations in Hamilton formulation at page 98 in [80].

The canonical momentum[3] p is correlated with the Lagrangian and the generalised coordinates as follows

$$p = \frac{\partial \mathcal{L}}{\partial \dot{q}} \qquad \text{and} \qquad \dot{q} = \frac{\partial \mathcal{H}}{\partial p} \quad (2.47)$$

therefore, the Hamiltonian is

$$\mathcal{H}(q, p, t) = \sum \dot{q} p - \mathcal{L}(q, \dot{q}, t). \quad (2.48)$$

If the Lagrangian comprises n second-order differential equations, the Hamiltonian consists of $2n$ first-order differential equations, known as the canonical equations. Let us explore how we can derive them. The derivation of Eq. (2.48) with respect to q and the time gives

$$\frac{\partial \mathcal{H}}{\partial q} = -\frac{\partial \mathcal{L}}{\partial q} \qquad \frac{\partial \mathcal{H}}{\partial t} = -\frac{\partial \mathcal{L}}{\partial t} \quad (2.49)$$

and from the Lagrange equation the following identity is verified

$$\dot{p} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) = \frac{\partial \mathcal{L}}{\partial q}. \quad (2.50)$$

Finally, the Hamilton's equations, also called canonical equations, are

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p} \tag{2.51}$$

$$\dot{p} = -\frac{\partial \mathcal{H}}{\partial q}. \tag{2.52}$$

In the book by Hamill 2013 [80], it is demonstrated that the Hamiltonian formulation is an explicit function of time, even if the Lagrangian is not. Hence, if the Lagrangian is not directly dependent on time and the potential energy solely relies on the coordinates, the Hamiltonian represents to the total energy of the system

$$\mathcal{H} = U_{kin} + U_{pot}. \tag{2.53}$$

2.2.3 Equation of motion

The equation of motion in SPH derives from the combination of the first law of thermodynamics and the second of Hamilton's equation. Thus, the internal energy per unit volume ($dV = -m/\rho^2\, d\rho$) is

$$dU = T\, dS + \frac{P}{\rho^2}\, d\rho\,. \tag{2.54}$$

Considering a particle b, the value of U at constant entropy it is a function of pressure P and density ρ:

$$\left.\frac{\partial U_b}{\partial \rho_b}\right|_S = \frac{P}{\rho^2} \tag{2.55}$$

whereas, the density gradient at particle b with respect to the coordinates of particle a reads

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \sum_c m_c \frac{\partial W_{bc}}{\partial \mathbf{r}_a}. \tag{2.56}$$

The second canonical equation with respect to the particle a is

$$\dot{p}_a = -\sum_b \frac{P_b}{\rho_b^2} \sum_c m_c \frac{\partial W_{bc}}{\partial \mathbf{r}_a} \tag{2.57}$$

and finally, we get the Euler-Lagrange equation of motion

$$\frac{D\mathbf{v}_a}{Dt} = -\sum_b m_b \left[\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right] \frac{\partial W_{ab}}{\partial \mathbf{r}_a}. \tag{2.58}$$

2.2.4 Total energy conservation

As a consequence of Eq. (2.53) we can demonstrate how the total energy E is conserved [153]. With respect to a particle a the Hamiltonian is:

$$\mathcal{H} = \sum_a \mathbf{v}_a \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{v}_a} - \mathcal{L} = \sum_a m_a \left(\frac{1}{2} v_a^2 + U_a \right) = E. \tag{2.59}$$

Therefore, the total energy derivative can be written as

$$\frac{dE}{dt} = \sum_a \frac{dE_a}{dt} = \sum_a m_a \left(\mathbf{v}_a \cdot \frac{D\mathbf{v}_a}{Dt} + \frac{dU_a}{dt} \right). \tag{2.60}$$

The last term in the RHS of Eq. 2.60 is defined by the internal energy, which can be derived from Eq. 2.55 and become

$$\frac{dU}{dt} = \sum_a \frac{dU_a}{dt} = \sum_a \frac{P_a}{\rho_a^2} \frac{d\rho_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla W_{ab}. \tag{2.61}$$

Finally, the total energy variation can be written as

$$\frac{dE}{dt} = -\sum_a \sum_b m_a m_b \left[\frac{P_a}{\rho_a^2} \mathbf{v}_b \cdot \nabla W_{ab} + \frac{P_b}{\rho_b^2} \mathbf{v}_a \cdot \nabla W_{ab} \right] = 0. \tag{2.62}$$

Since the two-fold summation of antisymmetric formulation gives zero, the conservation of total energy is confirmed.

2.2.5 Demonstration of density formulation

The density formulation in SPH is consistent with the continuity equation [153]. The time derivative of Eq. (2.39) is

$$\left\langle \frac{d\rho}{dt} \right\rangle_a = \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla W_{ab} \tag{2.63}$$

that can also be expressed as

$$\begin{aligned} \left\langle \frac{d\rho}{dt} \right\rangle_a &= \mathbf{v}_a \cdot \sum_b \frac{m_b}{\rho_b} \rho_b \nabla W_{ab} - \sum_b \frac{m_b}{\rho_b} (\rho_b \mathbf{v}_b) \cdot \nabla W_{ab} \\ &\approx \mathbf{v}_a \cdot \nabla \rho - \nabla \cdot (\rho \mathbf{v}) \approx -\rho_a (\nabla \cdot \mathbf{v})_a \end{aligned} \tag{2.64}$$

confirming that the density summation is a discrete SPH solution for the continuity equation.

2.2.6 Why the Hamiltonian formulation is preferable

Among the particle approaches, SPH is one of the few that uses the Hamiltonian formalism. While we have discussed the theoretical dif-

ferences between the two approaches, what are the practical benefits of using the Hamiltonian? Simply put, it helps in keeping the particle distribution uniform, but let us explore how.

In the work by Price 2012 [153], the author reproduces a box filled with particles and periodic boundary conditions. Initially, the particles are randomly distributed, and they have constant pressure values. In Eq. (2.58), the Hamiltonian equivalent of the Euler-Lagrange equation, the force derived for an isothermal system is solely determined by pressure. When assuming constant pressure, as in the test case by [153], the force does not reach zero but instead becomes negative, indicating repulsion between neighbouring particles. Essentially, this leads to the particles adjusting their positions randomly, forming a "glass-like" or, more precisely, a crystalline state.

As discussed in Section 2.1.8 on error analysis in SPH, maintaining a well-ordered distribution of particles is crucial for accuracy. Consequently, the inherent "re-meshing" capability of the Hamiltonian in SPH, which promotes a favourable arrangement of particles, is highly advantageous. Methods using Lagrangian formalism, where exact derivative computation is preferred over conservation, do not have any inherent re-meshing ability. In particle methods, particles tend to form a random disordered arrangement, necessitating complex re-meshing approaches. The same applies to Lagrangian methods with grids: manual re-meshing is necessary as the grid distorts over time.

The preference for antisymmetric formulations, as discussed in Section 2.1.6, is ultimately justified and stems from their alignment with the definition of the Hamiltonian. In mathematical terms, the particles will keep moving until this condition is satisfied

$$\sum_b m_b \left(\frac{1}{\rho_a^2} + \frac{1}{\rho_b^2} \nabla W_{ab} \right) \approx 0 \tag{2.65}$$

which corresponds to the minimum energy state, also known as Hamilton's principle. There is anyway a major drawback: the automatic particle re-arrangement is unresponsive to small perturbations, e.g., viscous effects of the boundary layers.

We must also acknowledge that the algorithm can hardly change without breaking some of the conservation properties given by the Hamiltonian formalism [153]. Indeed, one can only change the density formulation (the real core of our SPH method) or, alternatively, consider additional terms and constraints in the Lagrangian (e.g., implement a variable smoothing length algorithm).

2.2.7 Final remarks

In section 2.2.1 it was included a list of the most important features inherently related to SPH. The knowledge gained so far can help us to make the following points more clear [154, 153].

1. In SPH, the continuity equation is resolved exactly since the density summation formula is a derivation form of the continuity equation and it is time-independent thanks to the Hamiltonian conservation law.
2. The advection is done perfectly using the canonical equations.
3. The total energy conservation implies zero intrinsic dissipation.
4. The exact and simultaneous conservation of mass, momentum, angular momentum, and energy is a consequence of the application of the Hamiltonian.
5. Hamilton's principle guarantees the minimum energy state for particles.
6. The resolution follows the mass, meaning that an improved convergence is reachable when increasing the number of particles computed. In contrast, this decision results in simulations being more time-consuming. The recommended approach is to discretise the domain, implementing a number of particles decided based on the available computational resources.

2.3 Governing equations

This section delves into the SPH approximation process of the continuity and momentum equation, the pressure handling, and the time integration scheme.

The general form for the conservation laws is:

$$\text{Rate of change} = \text{Advection} + \text{Diffusion} + \text{Source}.$$

In SPH the momentum equation given by the Hamiltonian formalism lacks of diffusive term, so its contribution must be added explicitly. However, this is not surprising since, for closed systems, Hamilton's equations perfectly conserve the energy. The discretisation approach to the diffusing term (i.e. the viscous force) will be explained in detail in the next Chapter 3.

2.3.1 Discetised continuity equation

The mass conservation is expressed by the continuity equation

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}. \tag{2.66}$$

In Section 2.2.5 it has been demonstrated that the continuity in SPH can be expressed by the summation formula:

$$\langle \rho \rangle_a = \sum_b m_b W_{ab}. \tag{2.67}$$

Alternatively, in 2003 Colagrossi and Landrini [35] presented the following equation

$$\left\langle \frac{d\rho}{dt} \right\rangle_a = \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla W_{ab}. \tag{2.68}$$

Despite Eq. (2.67) is consistent with the basic ideas of SPH, smoothing out the density field and making it stable since it lowers the probability of particle clustering, it is not suitable for free-surface flows. In fact, the kernel interpolation close to the surface is truncated, and the stability is lost. Thereby, the second formulation, Eq. (2.68), comes to handy when studying free-surface problems because. It depends on the velocity field rather than on the particle's arrangement. Unfortunately, it does not guarantee the conservation, so the density field is more likely to be affected by numerical errors.

In 2006, Hu and Adams [87] suggested the use of the following variation of Eq. (2.67)

$$\langle \rho \rangle_a = m_a \sum_b W_{ab}. \tag{2.69}$$

With this adjustment, the mass of neighbouring particles is excluded from the summation, resulting in an improvement in density calculation when dealing with multiphase flows.

2.3.2 Discretised momentum equation

Based on the Hamiltonian formalism, the set of equations used to solve the momentum of the system is the following

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P + \mathbf{f}_\nu + \mathbf{f}_\mathrm{b} + \frac{1}{\rho}\mathbf{f}_{st} \tag{2.70}$$

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}. \tag{2.71}$$

where Eq. (2.70) calculated the particle velocity and Eq. (2.71) is the advection equation, giving the particle position.

The source contribution in Eq. (2.70) is represented by the pressure gradient ∇P and the body force $\mathbf{f}_\mathrm{b}$ terms. The antisymmetric-type formulation better approximates the pressure gradient (see Eq. (2.58)) because it ensures the conservation. In addition, Hu and Adams [87] presented an improved variation of the pressure gradient term replacing the density at the denominator by the measure of particle number density $\theta_a = \sum_b W_{ab}$,

resulting in the following

$$\left\langle \frac{\nabla P}{\rho} \right\rangle_a = \frac{1}{m_a} \sum_b \left(\frac{P_a}{\theta_a^2} - \frac{P_b}{\theta_b^2} \right) \nabla W_{ab}. \tag{2.72}$$

The quantity $\mathbf{f}_b$ normally represents forces like gravity and other external excitations. As for the viscous force $\mathbf{f}_v$, this term will be treated in depth in Chapter 3.

The term $\mathbf{f}_{st}$ denotes the surface tension force. The dynamics of the fluid-fluid interface is thoroughly explained in the 2011 contribution by Bush [22], which attributes the force to intermolecular attraction. In each phase, two distinct behaviours are observed (representation in Fig. 2.6): within the bulk, molecules attract each other uniformly in all directions, giving zero net force; however, at the interface, molecules are just partially surrounded by their phase, leading to unbalanced forces. When fluids are immiscible, a tension parallel to the interface works to prevent the molecules from leaving the phase they belong to: it acts as if the interface would be a stretching membrane. The maintenance of the surface is energetically costly, so the system tries to minimize the surface area. Brackbill et al. successfully proposed the mathematical model of

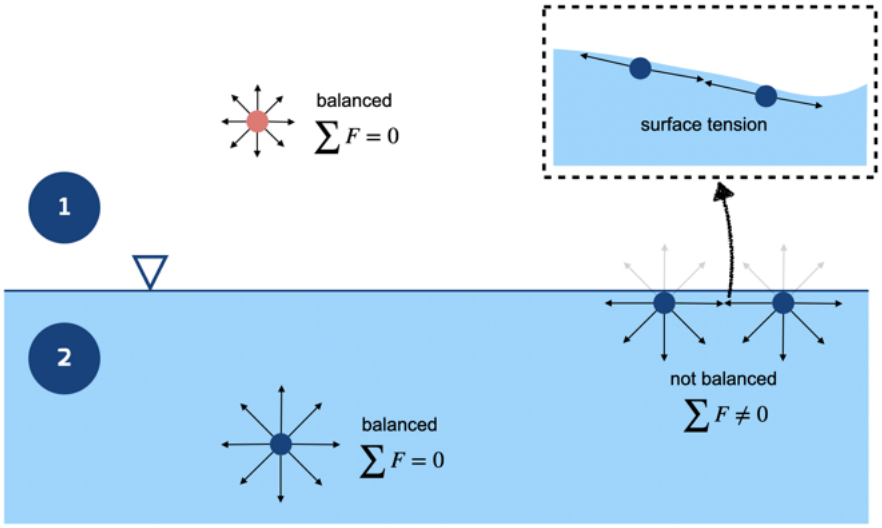


Figure 2.6: Surface tension representation: two phases denoted with tags (1) and (2), in the bulk of the each phase the molecules are subjected to net force equal zero $\sum F = 0$; whereas, near the interface this balance is not respected. The surface tension acts as a stretching membrane.

the surface tension in 1992 [19], and named it Continuum Surface Force method (CSF). It was only in 2000 that Morris [131] modified the original version for SPH. The surface tension uses the Young-Laplace equation, which defines the capillary pressure variation across the interface and relates the pressure with the surface curvature. The surface tension force per surface area can be written as

$$\mathbf{f}_{st} = \sigma_{st} \kappa \hat{\mathbf{n}} \tag{2.73}$$

where σ_{st} is the fluid property called surface tension coefficient, κ is the local curvature and $\hat{\mathbf{n}}$ is the unit vector normal to the interface. This vector is a function of the phase indicator c also called colour function,

which assumes an integer value per each phase; so $\hat{\mathbf{n}}$ can be written as the following

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|} = \frac{\nabla c}{|\nabla c|} \tag{2.74}$$

where the normal vector $\mathbf{n}$ for particle a is computed as

$$\mathbf{n}_a = \sum_b (\tilde{c}_a - \tilde{c}_b) \nabla_a W_{ab} V_b \tag{2.75}$$

only the values of $|\mathbf{n}_a|$ that exceed the threshold of $0.01/h$ are significant for calculation. The symbol $\tilde{c}$ represents the smoothed colour function, that is given by the following summation

$$\tilde{c}_a = \sum_b c_b W_{ab} V_b. \tag{2.76}$$

The local curvature can be written as

$$\kappa = -\nabla \cdot \hat{\mathbf{n}} \tag{2.77}$$

that is approximated in SPH as below

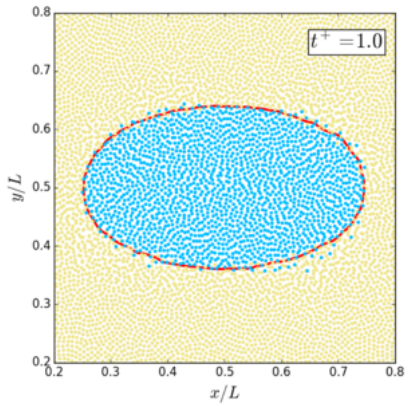
$$\kappa_a = \sum_b (\hat{\mathbf{n}}_a - \hat{\mathbf{n}}_b) \cdot \nabla W_{ab}. \tag{2.78}$$

There are cases where the phenomenon of micro-mixing arises. This phenomenon is characterised by a small number of particles from one phase traversing the interface and entering the bulk of the adjacent phase. We see an example in Fig. 2.7a depicting the oscillating drop benchmark: light blue particles of the drop phase leave the phase, cross the interface, and enter the external phase. These few particles alone cannot be interpreted as bubbles or droplets, so this phenomenon must be considered non-physical and avoided. Some remedies have been considered, starting with Colagrossi and Landrini in 2003 [35] and later in 2009 Grenier et al. [76]. An extra term is incorporated in the momentum equation to inhibit interface penetration. Among the various numerical adjustments proposed over the years, such as those outlined by Das and Das in 2009 [44], by Monaghan and Rafiee in 2013 [128] and the study by Zainali et al. in 2013 [201], we prefer to focus on the research conducted by Szewc et al. between 2013 and 2015 [185, 186]. In their work, the authors developed an interface correction aimed at enhancing the interface sharpness and avoiding micro-mixing. This correction manifests as as an additional force term on the RHS of the momentum equation and reads

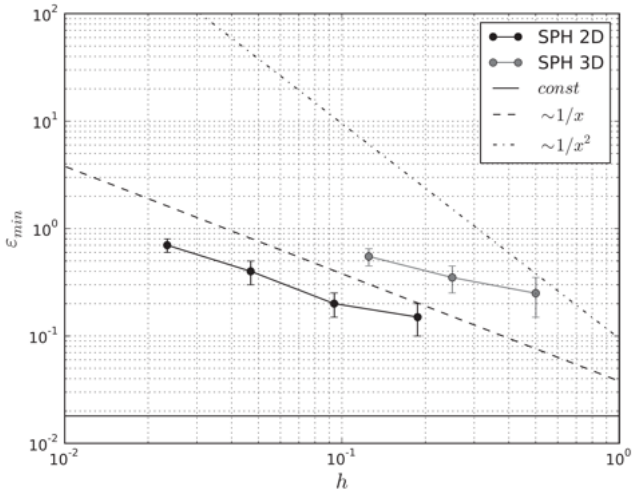
$$\Xi_a = \frac{\epsilon}{m_a} \sum_b \left(\frac{1}{\theta_a^2} + \frac{1}{\theta_b^2} \right) \tag{2.79}$$

where particles a and b belong to different phases. The interface sharpness correction coefficient ϵ has the dimension of pressure, its numerical value has been investigated in depth in [186], and in general, it should stay

within the range 0.1-0.5 as depicted in Fig. 2.7b.



(a) Micro-mixing effect in oscillating drop benchmark: particles belonging to the drop in blue colour leave the phase and enter the external phase. Picture from [140].



(b) Interface sharpness correction coefficient ϵ dependence on h in logarithmic scale. Plot from [186].

Figure 2.7: Micro-mixing issue: (a) example given by oscillating drop benchmark from [140]; (b) interface sharpness correction coefficient ϵ plot from [186].

In SPH, the implementation of turbulence models is still an open discussion. As pointed out in [151], the statistical turbulence modelling, such as RANS equations, is rather prohibitively for particle methods; they highly increase the complexity of the SPH approximation of the governing equations. The LES model is a more convenient option, the spacial filtering of turbulent small scales is somehow a similar idea to the smoothing approach given by the kernel function in SPH. Nevertheless, the LES model is a study in 3D scenarios and requires a minimum resolution to solve all the turbulent scales considered. Both these conditions can lead to extremely time-consuming SPH simulations, sometimes even impractical within nowadays machines. In our code and in the SPH applications presented in this manuscript, we do not address a turbulent model.

2.3.3 Pressure treatment and compressibility issue

The algorithm implemented in our code permits a slight degree of compressibility, where the density fluctuates according to the condition imposed on the Mach number, which must remain below the threshold of 0.1. This version is called Weakly Compressible SPH (WCSPH). Within this method, the governing equations also incorporate the Equation of State (EoS) for pressure, which behaves as a closing equation for the system, determining pressure based on density and ensuring the Mach number condition above.

We have tested both the Tait equation and the linear EoS that are respectively

$$P = \frac{s^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] \tag{2.80}$$

and

$$P = s^2(\rho - \rho_0). \tag{2.81}$$

In them, the symbol s denotes the numerical speed of sound, ρ_0 is the reference density (often the initial value) and γ stands for heat capacity ratio (that for incompressible liquids is $\gamma = 7$, while for compressible fluids, e.g. air, is $\gamma = 1.4$). Typically, the artificial speed of sound is set to be approximately one order of magnitude greater than the maximum velocity attained within the domain. This is done to ensure that pressure variations remain below 1%.

As explained in the 2018 study by Kajzer and Pozorski [93], in incompressible flows, the Mach number is considered to be zero, implying that the speed of sound approaches infinity, density variation is negligible, and the velocity field is divergence-free. Consequently, it is not feasible to establish a direct relationship between density and pressure, rendering the Equation of State (EoS) inapplicable. In a three-dimensional system, the Navier-Stokes equations consist of four, yet the unknown quantities total five (density, velocity vector, and pressure). Therefore, determining the pressure value is necessary for obtaining a solution. By combining the divergence of the momentum equation with the constraint of the continuity equation, one can derive the Pressure Poisson Equation (PPE). The PPE takes on an elliptic form, and knowledge (or estimation) of pressure values at the boundary is required to compute the entire fluid domain. For compressible or weakly compressible approaches, the mass conservation is a transport equation for density, and the pressure equation comes from the thermodynamic relation (e.g., ideal gas law); consequently, the partial differential for momentum equation becomes a hyperbolic or parabolic type.

Why choose WCSPH over an incompressible approach? Our team has already investigated the differences between Truly Incompressible SPH (ISPH) and WCSPH in the 2013 study [182], which was subsequently summarised in [140]. Let us briefly recall the features of the ISPH approach:

- the time step allowed is higher than in WCSPH,
- the numerical complexity is increased by the equation form which is elliptic-parabolic,
- the PPE-based algorithm is hardly parallelisable,
- this version of SPH is more sensible to errors due to the particle clustering effect; even though the particle position can be changed or re-initialized, this kind of application undermines the inherent advantages of SPH and, in multiphase flows, can also modify the interface.

and the ones of the WCSPH approach:

- acoustic effects,
- time step restrictions based on the value of the speed of sound;

Considering these characteristics, our team has mostly opted for the WCSPH.

2.3.4 Time integration

In numerical simulations, continuum integrals are converted into sums of discretised quantities. When modelling motion within a specified time frame, the equations are integrated over time using incremental time steps. The time step quantity is denoted by Δt and divides the analysis into small consecutive steps. Selecting an appropriate value for Δt is no trivial matter; indeed, ensuring a sufficiently small time step is vital for preserving system stability and achieving precise outcomes. Nonetheless, decreasing Δt results in longer computational durations, prolonging the time it takes to run simulations.

As shown in [182, 140], the algorithm implemented in our code is suitable for the first order Euler scheme, where the maximum threshold of Δt is given by the relation

$$\Delta t \leq \min\{\Delta t_{CFL}, \Delta t_{\nu}, \Delta t_{CSF}\}. \tag{2.82}$$

The time step corresponds to the minimum value among all three criteria. The first and the oldest criterion is the Courant-Friedrichs-Lewy (CFL), indicated by Δt_{CFL}: dates back to the 1928 research by Courant et al. [39] and it was designed for explicit time integration, it is also suitable for methods that model convection or wave phenomena. Its formulation for weakly compressible approaches is

$$\Delta t \leq \Delta t_{CFL} = 0.25 \frac{h}{s + v_{max}} \tag{2.83}$$

where v_{max} is the maximum resulting in the velocity field. For truly incompressible approaches, the numerical speed of sound s is neglected (considering PPE is used, the acoustic effects do not occur), thereby it becomes:

$$\Delta t \leq \Delta t_{CFL} = 0.25 \frac{h}{v_{max}}. \tag{2.84}$$

The Δt_{CFL} can be advantageous when dealing with ISPH, characterised by larger values, it can speed up the simulations. While opting for WCSPH in conjunction with the Euler scheme may appear unfavourable at first glance, it is in fact beneficial because it facilitated the parallelised computation of the code. The second criterion was proposed by Morris in 1997 [132], the aim was to handle explicit schemes for viscous flows.

This formulation is written as

$$\Delta t \leq \Delta t_\nu = 0.125 \frac{h^2}{\nu} \tag{2.85}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. The last condition comes from the work of Brackbill et al. first published in 1992 [19], it depends on the surface tension force CSF model, which by the way was also presented in the same paper. The formula is

$$\Delta t \leq \Delta t_{CSF} = \left(\frac{h^3 \min\{\rho_1, \rho_2\}}{2\pi\sigma} \right)^{1/2}. \tag{2.86}$$

2.4 Boundary conditions

In simulations, boundary conditions (BC) play a vital role in defining the limits and constraints of the computational domain. Moreover, there are numerous reasons justifying the necessity of BC in the majority of simulations. Generally, BC assist simulations in mirroring actual physical or environmental constraints, accommodating external influences or limitations on the system, and making the simulation computationally feasible. Thanks to BC, a wide array of scenarios can be explored within the simulation. In particular, in SPH, BC are crucial as they prevent non-physical behaviour of the system. The borders of the domain are critical areas since they cause kernel truncation; computing a half-full kernel results in infeasible outcomes. Additionally, BC may act as a barrier preventing particles from leaving the domain.

According to Crespo 2008 [40], there are three principal types of boundary conditions for SPH; these are listed below.

1. Repulsive particles: from an idea of Monaghan 1994 [123], this approach apply central forces on the fluid particles. Similarly to molecule behaviour, when a frontier and a fluid particle are at a distance r the attraction is determined by Lennard-Jones potential.
2. Ghost particles: when a real particle from inside the domain gets closer to the boundary, a mirrored fictitious particle is created that has same reflected properties as the parent particle (see Fig. 2.8). The properties of the mirror (or ghost) particles, such as mass, velocity, and pressure, are specifically assigned to reflect the macroscopic boundary conditions imposed on the fluid. The ghost particles are not dynamically followed in time but enter the summation formulae to compute the evolution of the true (parent) particles representing the fluid in the boundary region. Based on an idea of Libersky et al. in 1993 [104], it was officially presented by Morris et al. [132], and further developed by other researchers such as Cummins and Rudman (1999) [42], Pozorski and Wawreńczuk

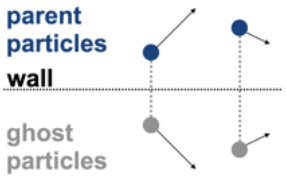


Figure 2.8: Ghost particles are mirrored and fictitious boundary particles.

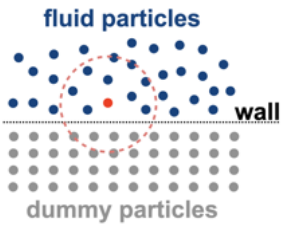


Figure 2.9: Dummy particles are wall particles, they are not advected resulting in being stationary.

(2002) [152], Adami et al. (2012) [1], Ferrand et al. (2013) [64] and many others. This method is relatively popular.

3. Dummy particles: these particles play a role in calculating fluid continuity and momentum, similar to fluid particles. Unlike fluid particles, however, they remain stationary and are not advected (see Fig. 2.9). This approach, also known as Dynamic Boundary Condition (DBC) in the Spheric community, is highly popular due to its simplicity and adaptability, particularly for domains with complex geometry. It was first implemented by Dalrymple and Knio [43], but many improved versions have been developed later. Although it has been called DBC for years, it has also been recently referred to as the Fixed Ghost Particles condition (GPBC) [1].

Our code adopts the ghost particles approach, with a comprehensive explanation of the algorithm and other programming features provided in Appendix C at this manuscript's end.

New viscous force validation and rising bubble benchmark analysis

3

In Chapter 2 we established that the Hamiltonian formulation stands apart by maintaining zero dissipation, a characteristic distinct from Eulerian schemes where diffusion naturally arises from discretization. However, in SPH, to uphold the conservation of momentum, both linear and angular, the dissipative term is introduced explicitly.

In this chapter we give an overview of the conventional formulations for viscous force term in SPH. Then, we introduce our novel formulation mVER, a variation of the formula presented for multiphase flows in the book[1] by Violeau [192]. The mVER formulation is validated against two benchmarks: the Poiseuille flow and the rising bubble. Specifically, we delve into the rising bubble test, taking this opportunity to explore acoustic effects (mentioned already in Section 2.3.3), particularly evident in this case.

1: We refer to Eq. 6.59 at page 404 of the book [192].

3.1 Viscous term in the momentum equation

The Cauchy stress tensor, denoted by σ, describes the distribution of stresses within the fluid at a particular point. The stress tensor for Newtonian fluid is expressed as the sum of the pressure part, where the normal stresses are given by: the hydrostatic pressure p (and $\mathbf{I}$ that is the identity tensor), plus the viscous part $\boldsymbol{\tau}$. The formulation is then written as in the following

$$\sigma = -p\mathbf{I} + \boldsymbol{\tau}.$$

The viscous stress $\boldsymbol{\tau}$ is the measure of the force parallel to a surface; when considering the fluid composed by layers, it is the quantity that causes one layer to slide or deform relatively to the adjacent layer. Since in the present work we implement Newtonian fluids exclusively, the shear stress is a function of the rate-of-deformation tensor (or equivalently said the velocity gradient) and of the thermodynamics of the system.

As pointed out in Colagrossi et al. (2017) [37], nowadays implementations of incompressible flows are also possible using methods that allow for slight compressibility, and WCSPH is one of them.

For compressible Newtonian fluids, the viscous stress that contributes to the Navier-Stokes equation is

$$\boldsymbol{\tau} = \mu\big(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\big) + \lambda(\nabla\cdot\mathbf{v})\mathbf{I}. \tag{3.1}$$

and in the tensorial form it reads

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \delta_{ij} \lambda \frac{\partial v_k}{\partial x_k} \tag{3.2}$$

where μ is the shear dynamic viscosity, δ_{ij} the Kronecker delta and λ is the so-called second viscosity (or bulk viscosity coefficient). The coefficients μ and λ can be related to give the bulk viscosity ζ. If μ represents the coefficient used to quantify the resistance to shear, the bulk viscosity ζ, also known as volume viscosity, gauges the extent of the energy dissipated during compression or expansion of the fluid. In other words, the latter coefficient serves as an indicator that the thermodynamic equilibrium of the system has been perturbed. The Stokes hypothesis suggests that the bulk viscosity can be assumed to be zero, mainly for incompressible Newtonian fluids where the density changes associated with compression or expansion are minimal. While this simplification works well for many Newtonian fluids in various practical scenarios, it might not capture more complex behaviours exhibited by non-Newtonian fluids or under certain extreme conditions. Besides, early researches on the topic already questioned the hypothesis, claiming that the value of λ can be of the order of magnitude of μ for realistic fluids, thus obviously not negligible. An interesting point of view is presented in the paper by Colagrossi et al. [37], where the validity of the Stokes hypothesis in the SPH model is discussed. The critical idea stated in [37] is that the Stokes hypothesis contradicts the conservation of the angular and linear momentum; therefore if the bulk viscosity is not null, the second viscosity is equal to the shear viscosity.

The viscous force $\mathbf{f}_\nu$ in the momentum equation, Eq. (2.70), is the relation

$$\mathbf{f}_\nu = \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} = \frac{1}{\rho} \nabla \cdot \left[\mu (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \right) + \lambda (\nabla \cdot \mathbf{v}) \mathbf{I} \Big]; \tag{3.3}$$

which is given by the second order derivative of the velocity that in SPH is yielded by first order derivative of the kernel.

In SPH, there are two main methods for dealing with the force $\mathbf{f}_\nu$. Apart from trying to approximate directly the physical viscosity through a laminar viscosity model, the alternative is the implementation of the artificial viscosity (AV). In general, the AV is a numerical tool that is different from zero and it is defined as a continuous term whenever the fluid undergoes a compression, thereby acting as a repelling term avoiding for particles gathering[2]. It vanishes in case of expansion. The AV satisfies a list of conditions meanwhile the laminar viscosity models are requested to be consistent with the physical formulation. *Our ambition is to propose a new viscous force term that meets the latter demand, we will call it the modified Violeau, Español and Revenga (mVER) formulation.*

2: Further details in the web page about Artificial Viscosity [47] where are referenced: the 13^{th} chapter of the *Handbook of Numerical Analysis* by Loubère et al. [109] that include in page 21 a long list of requirements for AV in Finite Volume schemes; the 16^{th} chapter of *An Invitation to Applied Mathematics* authored by Chicone [28].

3.1.1 The most common formulations for viscous force

The artificial dissipation term by Monaghan [124] is the most common formulation

$$(\mathbf{f}_v^{AV})_a = \sum_b m_b \Pi_{ab} \nabla W_{ab} \tag{3.4}$$

denoted by Π_{ab}, the term reads

$$\Pi_{ab} = \begin{cases} -\alpha \dfrac{h\overline{s}_{ab}\mu_{ab}^{AV}}{\overline{\rho}_{ab}} & \text{if}(\mathbf{v}_a - \mathbf{v}_b)\cdot(\mathbf{r}_a - \mathbf{r}_b) < 0 \\ 0 & otherwise \end{cases} \tag{3.5}$$

where

$$\mu_{ab}^{AV} = \frac{(\mathbf{v}_a - \mathbf{v}_b)\cdot(\mathbf{r}_a - \mathbf{r}_b)}{(\mathbf{r}_a - \mathbf{r}_b)^2 + \epsilon^2} \tag{3.6}$$

is the artificial viscosity μ_{ab}^{AV}. The quantities with the overbar refer to the average over a particle pair (e.g. for the speed of sound we have the mean value $\overline{s}_{ab} = \frac{1}{2}(s_a + s_b)$) and α is a dimensionless parameter. The ϵ is necessary to avoid zero value in the denominator, and usually it is set to be $0.01h^2$. The AV matches three important requirements [153]:

- it is Galilean invariant;
- it vanishes for rigid body rotation;
- it conserves the total linear and angular momentum.

Moreover, as stated by Price in 2011 [154], the viscous effect can be considered as a combination of bulk and shear viscosity, whereas numerical parameters as α in Eq. 3.5 can be determined analytically from theoretical physics (see Hu and Adams 2006 [88]).

In the work by Morris et al. of 1997 [132], the authors stated that, although the AV by Monaghan was very common, it was not able to produce accurate velocity profiles for their simulations. For this reason, they suggested a novel viscous term to mimic physical viscosity, formulated as follows:

$$(\mathbf{f}_v^{M})_a = \sum_b \frac{m_b(\mu_a + \mu_b)\mathbf{r}_{ab}\cdot\nabla W_{ab}}{\rho_a\rho_b(\mathbf{r}_{ab}^2 + 0.01h^2)}\mathbf{v}_{ab}. \tag{3.7}$$

Unfortunately, the above formulation conserves exactly only the linear momentum while approximating the angular momentum. The conservation of the angular momentum is fundamental to ensure SPH condition for which particles cannot rotate. The Stokes hypothesis is taken into account, and it is assumed that the bulk viscosity is zero.

Hu and Adams introduced an advanced function in 2006 [87] designed for multiphase flows, with the phases denoted by the superscripts l and k. If a is a particle of the fluid phase k and b a particle that belongs to

phase l, the equation reads

$$(\mathbf{f}_v^{HA})_a = \frac{1}{m_a}\sum_b \frac{2\mu^k\mu^l}{\mu^k+\mu^l}\left(\frac{1}{\theta_a^2+\theta_b^2}\right)\frac{\mathbf{r}_{ab}\cdot\nabla W_{ab}}{\mathbf{r}_{ab}^2+0.01h^2}\mathbf{v}_{ab} \tag{3.8}$$

where the particle number density θ has been added. The average value of viscosity $(2\mu^k\mu^l)/(\mu^k+\mu^l)$ was probably inspired by the 1998 publication by Cleary [31], in which a modified version of the Monaghan's AV was used. Later in 2013, the average viscosity will be named by Adami et al. [2] the inter-particle-averaged shear viscosity.

The two types of equations, the first type is represented by AV and the second type includes Morris and Hu and Adams, have been smartly compared in the manuscript by Green in 2017 [74], and we agree on saying that the AV formulation preserves the linear and angular momentum, while $\mathbf{f}_v^{HA}$ is a finest approximation of the velocity Laplacian (because it also keeps the velocity direction). Until the research of the present manuscript was conducted, in our code the $\mathbf{f}_v^{HA}$ equation was preferred and implemented with satisfactory outcomes.

3.1.2 SPH approximation of the viscous force term

Even though both $\mathbf{f}_v^{M}$ and $\mathbf{f}_v^{HA}$ perform efficiently, they are an incomplete compromise of a theoretically consistent formulation of the viscous force.

Lately, our focus has been primarily directed towards the version in the textbook by Violeau [192], which determines more effectively the variability of viscosity across the interface, especially for multiphase flows. The formulation is the following:

$$(\mathbf{f}_v^{V})_a = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\left[\frac{\overline{\mu}_{ab}^{Arith}}{r_{ab}}(d+2)(\mathbf{u}_{ab}\cdot\mathbf{e}_{ab})\mathbf{e}_{ab}\frac{\partial W}{\partial r_{ab}} + \frac{\overline{\mu}_{ab}^{Arith}}{r_{ab}}\mathbf{u}_{ab}\frac{\partial W}{\partial r_{ab}}\right]. \tag{3.9}$$

where $\overline{\mu}_{ab}^{Arith} = (\mu_a + \mu_b)/2$ is the arithmetic mean of the particles viscosities. This formula is built on the Laplacian proposed by Español and Revenga in 2003 [58] (see section 2.1.7).

In the work by Grenier et al. (2013) [77] the parameter $\overline{\mu}_{ab}$ has been referred to as inter-particle dynamic viscosity (while in the previous section it has been called the inter-particle-averaged shear viscosity) and has been proved to better perform when it is formulated by the harmonic mean rather than the arithmetic. The authors have tested a two-fluid Poiseuille flow with the Morris formula $\mathbf{f}_v^{M}$, in which in one phase the dynamic viscosity μ is two orders of magnitude greater than in the other phase. We see in Fig. 3.1 the error at the interface that greatly increases for the simulation that computes the arithmetic mean. The

authors concluded that the harmonic mean has a better smearing action, it helps to avoid abrupt viscosity changes across the interface.

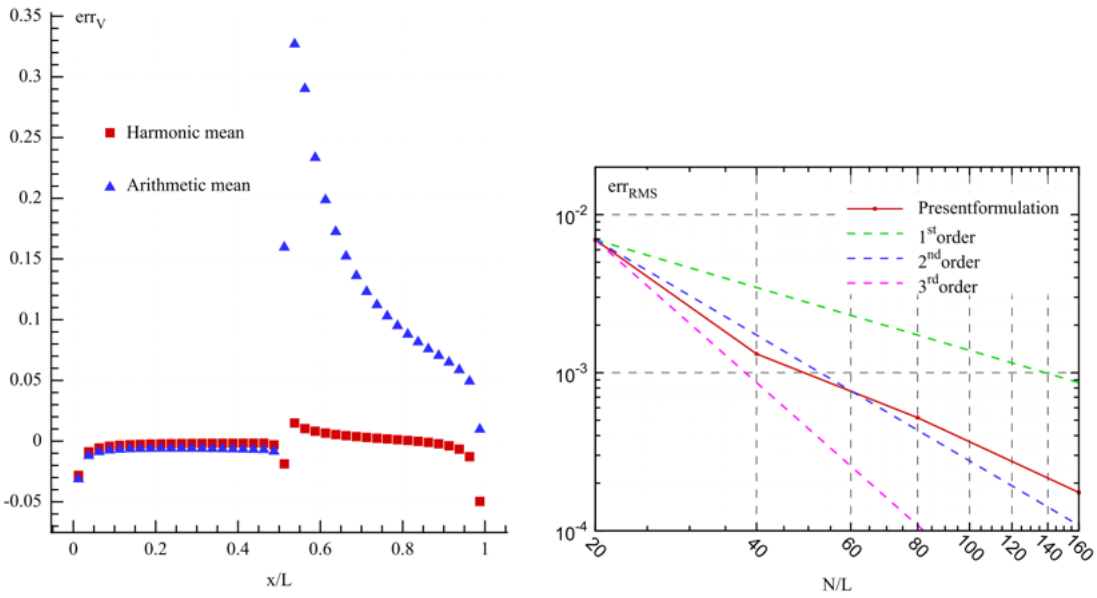


Figure 3.1: Graphs related to simulation of one-fluid Poiseuille flow by [77], comparison between arithmetic and harmonic mean for inter-particle viscosity. On the left: display of the relative error at $t = 10$s. On the right: the convergence study, L is the channel width and N is the number of particles along L.

Inspired by all these works, we have implemented in our code the Violeau formulation $\mathbf{f}_\nu^V$, replacing, as a novelty, the $\overline{\mu}_{ab}^{Arith}$ with the $\overline{\mu}_{ab}^{Harm}$ (by [77]):

$$\overline{\mu}_{ab}^{Harm} = \frac{2\mu_a\mu_b}{\mu_a + \mu_b}. \tag{3.10}$$

Finally, the proposed formulation reads

$$(\mathbf{f}_\nu^{mVER})_a = \frac{1}{\rho_a}\sum_b \frac{m_b}{\rho_b}\left[\frac{\overline{\mu}_{ab}^{Harm}}{r_{ab}}(d+2)(\mathbf{u}_{ab}\cdot\mathbf{e}_{ab})\mathbf{e}_{ab}\frac{\partial W}{\partial r_{ab}} + \frac{\overline{\mu}_{ab}^{Harm}}{r_{ab}}\mathbf{u}_{ab}\frac{\partial W}{\partial r_{ab}}\right]. \tag{3.11}$$

Our goal is to define a proper SPH expression for the viscous force: this new formulation is in accordance with the Violeau equation where the kernel Laplacian applied is the Español and Revenga [58] one. For this reason, the acronym of our proposal is mVER that stands for the modified Violeau Español and Revenga formulation $\mathbf{f}_\nu^{mVER}$.

3.2 Validation of the new mVER viscous force term: steady-state Poiseuille flow

We validate the mVER force in Eq. (3.9) by conducting a comparative analysis of the three formulations: the Hu and Adams one $\mathbf{f}_\nu^{HA}$, the Violeau one $\mathbf{f}_\nu^V$ and the mVER one $\mathbf{f}_\nu^{mVER}$. This evaluation is conducted within the framework of Poiseuille benchmark, in single-phase and

two-phase flow scenarios. One characterisation of this benchmark is the absence of gravity effect and external forces, thus this investigation allows us to focus on the viscous force and to demonstrate the effectiveness and applicability of the mVER formulation.

3.2.1 Single-fluid Poiseuille flow

The single-fluid Poiseuille flow in plane channel is characterized by the parameters: the channel width is $L = 1\,\mathrm{m}$, density and viscosity of the fluid are respectively $\rho_0 = 1\,\mathrm{kg/m^3}$ $\mu_0 = 1\,\mathrm{Pa}\cdot\mathrm{s}$ and the acceleration on x-direction $\mathrm{f}_b = 8\,\mathrm{m}\cdot\mathrm{s}^{-2}$. The maximum velocity characterizing the fluid is $v_0 = 1\,\mathrm{m/s}$ [195].

In Fig.3.2 the analytical solution is represented by a solid light-blue line while scatter points show SPH results. We compare three sets of outcomes, each of them from a different viscous force formulation: the data points in red are obtained with Eq. (3.8) by [87]; the green crosses are results from Eq. (3.9) with arithmetic mean by [192]; finally cyan points stand for the mVER formulation Eq. 3.9 with harmonic mean Eq. (3.10).

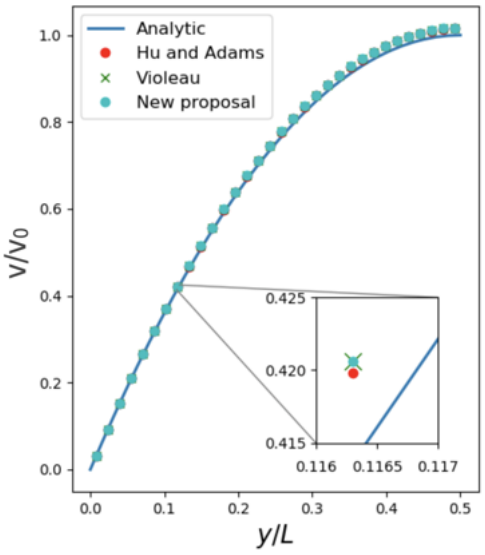


Figure 3.2: Steady-state Poiseuille flow in plane channel: half-channel single-fluid configuration.

All the SPH sets slightly over-predict the analytic solution and give almost the same results, indeed the data are very close one to each other (see close-up in the figure). We can convene that for single-fluid configuration, any formula taken into consideration properly performs the viscous effect.

3.2.2 Two-fluid Poiseuille flow

The Poiseuille flow in Fig. 3.3 features two immiscible phases; the interface overlaps the center line of the channel, which has width equal to $L = 1\,\mathrm{m}$. The phase parameters are: $\rho_0^I = \rho_0^{II} = 1\,\mathrm{kg/m^3}$, $\mu_0^I = 10\,\mathrm{Pa}\cdot\mathrm{s}$

and $\mu_0^{II} = 0.1\,\mathrm{Pa}\cdot\mathrm{s}$, with x-coordinate $\mathrm{f}_b = 1\,\mathrm{m}\cdot\mathrm{s}^{-2}$. The phase I is located on the lower part of the channel $0.0 \leq y \leq L/2$, the second phase II in-between the half of the channel and the upper wall $L/2 \leq y \leq 1.0$. As for the previous validating case, we compare the three formulations: the data points in red are obtained with Eq. (3.8) by [87]; the green points result from Eq.(3.9) with arithmetic mean by [192]; cyan points stand for the new proposal mVER, hence Eq. (3.9) with the harmonic mean in Eq. (3.10).

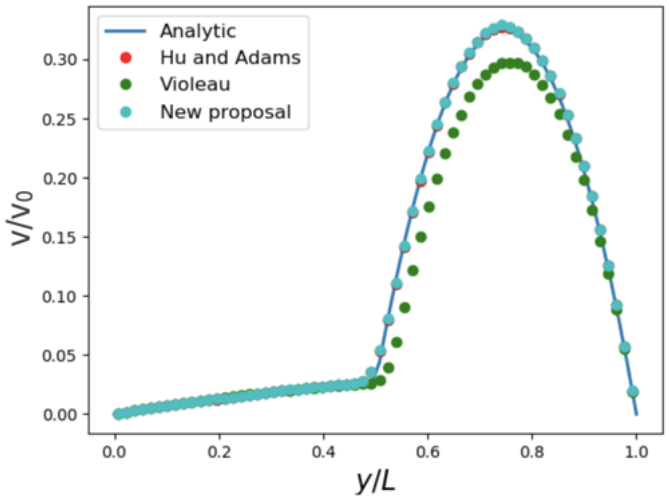


Figure 3.3: Steady-state Poiseuille flow in plane channel: two-fluid configuration.

The analytical solutions for each phase are respectively (for reference [12]):

$$v_y^I = \frac{\mathrm{f}_b L^2}{2\mu_0^I}\left[\left(\frac{2\mu_0^I}{\mu_0^I+\mu_0^{II}}\right)+\left(\frac{\mu_0^I-\mu_0^{II}}{\mu_0^I+\mu_0^{II}}\right)\left(\frac{x}{L}\right)-\left(\frac{x}{L}\right)^2\right] \tag{3.12}$$

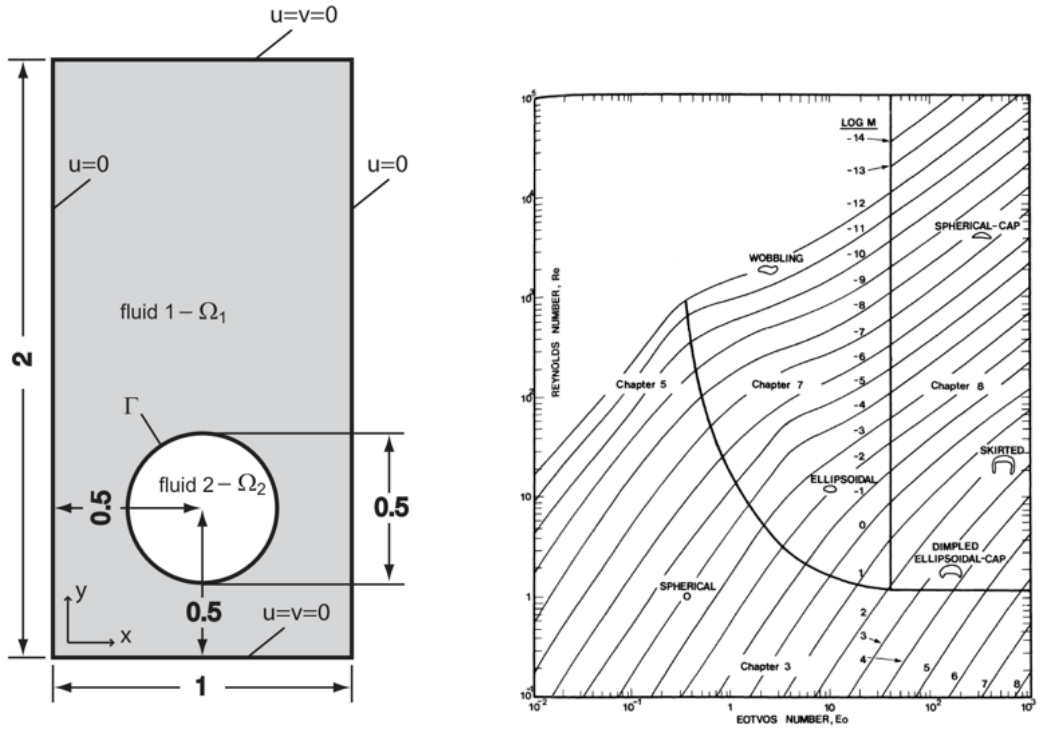
$$v_y^{II} = \frac{\mathrm{f}_b L^2}{2\mu_0^{II}}\left[\left(\frac{2\mu_0^{II}}{\mu_0^I+\mu_0^{II}}\right)+\left(\frac{\mu_0^I-\mu_0^{II}}{\mu_0^I+\mu_0^{II}}\right)\left(\frac{x}{L}\right)-\left(\frac{x}{L}\right)^2\right]. \tag{3.13}$$

The SPH results data for the phase I, with the highest dynamic viscosity, do not differ much from analytic formulation. The second phase II, which is the one with lower viscosity, is well performed by [87] and by the mVER formulation, but the data obtained with the Violeau model feature lower velocity. This behaviour confirms the idea that the harmonic mean for the viscosity coefficient is a better choice than the arithmetic one, especially for multiphase flows. The mVER formulation and the Hu and Adams viscous force give comparable results; both are good options in case of single- and two-phase Poiseuille flows, and it is hard to define which formulation perform better. A deeper investigation on their differences would be a possible idea for future work, testing these formula in other benchmarks and possibly in three-phases applications.

3.3 Rising bubble benchmark

The second validation benchmark involves the rising bubble study case. In general, bubbly flows pose challenges in numerics and have the potential to unveil the model's ability to handle multiphase flows with significant density and viscosity differences, especially when the interface undergoes deformation.

We are replicating the study conducted by Hysing et al. [91] (2009), in which are presented two different regimes of two-dimensional bubble rising in a column of denser fluid. This benchmark is notable and has been discussed in the CFD community (including SPH) for some time.



(a) Initial configuration and boundary conditions

(b) Bubbles regimes depending on Reynolds and Eötvös numbers

Figure 3.4: Rising bubble benchmark: (a) domain size in space units, initial configuration of the bubble and boundary conditions are included (from [91]); (b) the graphic shows how the bubble shape can be determined on the basis of the Reynolds and Eötvös numbers (from the book by Clift et al.[32] (1978).

The simulated domain arrangement is illustrated in Fig. 3.4a: in a column of fluid, which is periodic in the x-direction and bounded by walls along the y-direction, at time zero one bubble is located above the bottom wall. The gravity pushes the bubble upwards, while the shear effect and the non-uniformity of hydrostatic pressure at the surface cause the bubble deformation.

The simulation stops after 3 s, the domain is 1 m wide and 2 m high while the radius of the bubble is 0.25 m and more properties are listed in Tab.

3.1, where the first and second phase are denoted by the subscripts I and II, respectively.

Table 3.1: The main parameters of [91] benchmark, included both first and second cases.

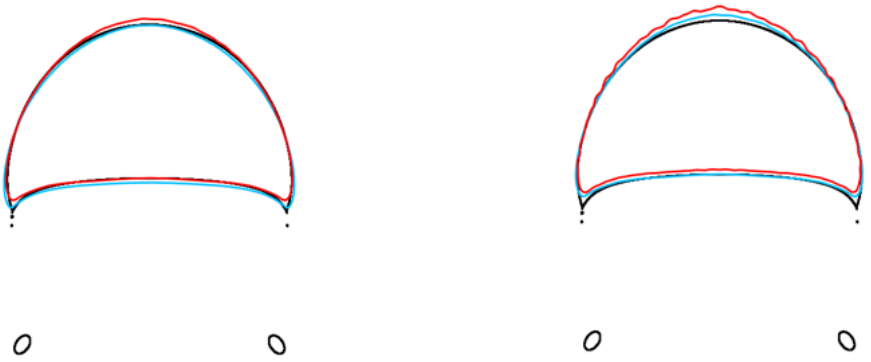
Case	$\rho_0^I\ [kg/m^3]$	$\rho_0^{II}\ [kg/m^3]$	$\mu_0^I\ [Pa \cdot s]$	$\mu_0^{II}\ [Pa \cdot s]$	$g\ [m/s^2]$	$\sigma_{st}\ [N/m]$	Re	Eo
1^{st}	1000	100	10	1.0	0.98	24.5	35	10
2^{nd}	1000	1	10	0.1	0.98	1.96	35	125

In addition to density and viscosity coefficients of each fluid, another physical quantity fundamental to characterize the flow is the surface tension coefficient σ_{st}. Using these variables, dimensionless parameters are built: the dimensionless Reynolds (Re) and Eötvös (Eo) numbers, of which formulae are:

$$Re = \frac{\rho_0 U_g L}{\mu_0}, \qquad Eo = \frac{\rho_0 U_g L}{\sigma}$$

where $v_0 = \sqrt{2gr_0}$ is the gravitational velocity given by the bubble initial radius r_0 and $L = 2r_0$ is the length scale. The Re and Eo are functional to predict the bubble deformation, as observable from the graph in Fig. 3.4b (by Clift et al. [32]).
In cases of higher shear, the upper section of the bubble retains its spherical shape, while the lower portion deforms in a thin, extended layer resembling a skirt. Occasionally, this skirt becomes so thin that it breaks, creating a trail of smaller bubbles in its wake.

For simplicity, the pressure is calculated using the isothermal Eq. (2.81), where the heat capacity ratio is equal to one in both phases. The speed of sound is initialised to maintain the condition of weak compressibility; for safety's sake, we have opted for $s_0 = 15\,\mathrm{m/s}$ in the liquid phase.



(a) Resolution comparison

(b) Comparison with and without the Interface Correction factor

Figure 3.5: (a) Resolution test: the black line is the benchmark [91]; the red line is the lower resolution; the light blue the higher. (b) Interface Correction test: the black line is the benchmark [91]; the red line is without the correction; the light blue is with the correction, showing more regular shape.

Higher resolution typically leads to more precise results, but also entails a greater number of particles and the number of iterations required.

Therefore, the simulation time inevitably increases. In order to determine the optimal resolution for the simulation, we compared the outcomes of two resolutions:

1. the lower resolution has smoothing length $h = 0.015625$ m, the total number of particles is 32768, the number of particles composing the bubble phase is 3228;
2. the higher resolution has smoothing length $h = 0.0078125$ m, the total number of particles is 131072, the number of particles composing the bubble phase is 12892.

Upon comparing the degree of circularity[3] of both options with the benchmark, the root mean square error is 9.3% for the lower resolution, while for higher resolution 7.6%. Even though the higher resolution requires twice the simulation time, the results exhibit a discreet improvement, we can see the difference in bubble shape in Fig 3.5a: the black line is the benchmark, the red line denotes the lower resolution while the light blue the higher.

3: In [91] the circularity is defined as the ratio where the numerator is the perimeter of a circle that has equivalent to the bubble and the denominator actual bubble perimeter, which is calculated with the formula for the interface length in [142].

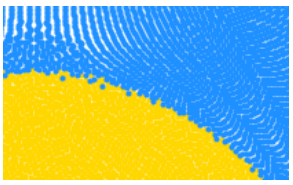
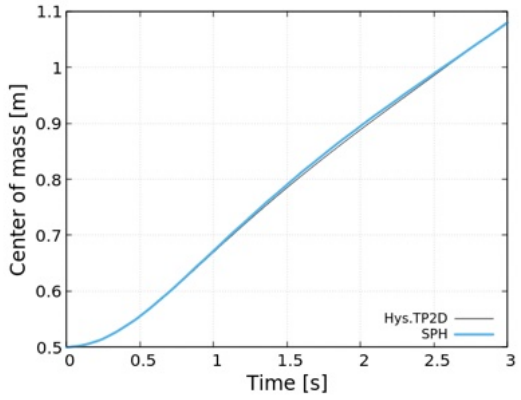


Figure 3.6: Example of micro-mixing effect in the rising bubble problem; particles of blue phase penetrating in the adjacent yellow phase.

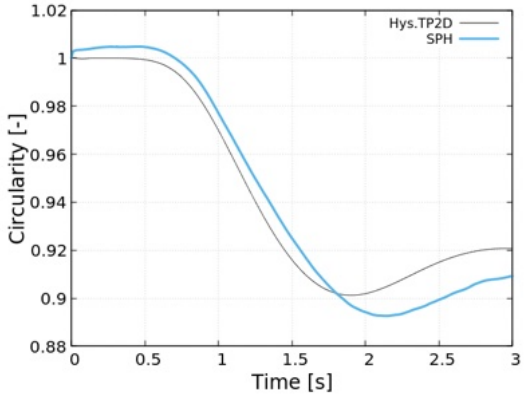
Moreover, we observed the presence of the micro-mixing effect, especially visible in the skirted regime case: in Fig. 3.6 some particles of the heavier phase, coloured in blue, penetrate the lighter phase, in yellow colour. The results are then enhanced by applying the Interface Correction (in section 2.3.2): the factor is dependent on the relative particle position. The factor should be of the order of 10 for both the lower and the higher resolution. After few trials, we determined that 25 was the accurate value. In Fig. 3.5b we see the comparison among bubble shapes obtained with the lower resolution: the black line is the benchmark, the red line denotes the simulation without the correction, the light blue represents the results with the correction showing clearly a more regular shape.

3.3.1 First case: ellipsoidal bubble

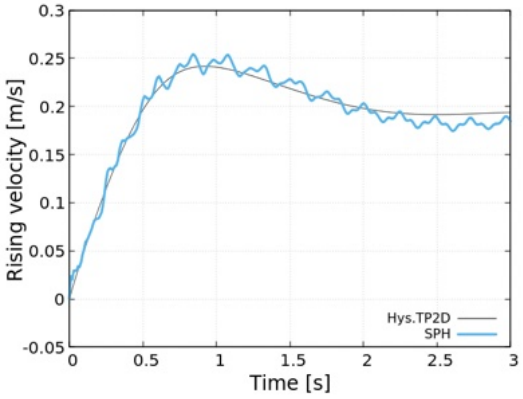
The first test case is characterised by domain features for which the final shape of the bubble is ellipsoidal, in Tab. 3.1 we see the parameters' values. In particular, the density ratio is $\rho_0^I/\rho_0^{II} = 10$ and the viscosity ratio is $\mu_0^I/\mu_0^{II} = 10$, $Eo = 10$. In time, the bubble will change from being spherical to ellipsoidal, flattened on the lower side, but we do not expect to see the formation of any skirt.



(a) Center of mass



(b) Circularity



(c) Rising velocity

(d) Shape at time 3s

Figure 3.7: Ellipsoidal regime, verification of results: the benchmark data [91] are denoted by the label Hys.TP2D (Transport Phenomena in 2D) and represented by a solid black line; SPH results are represented by solid light blue line.

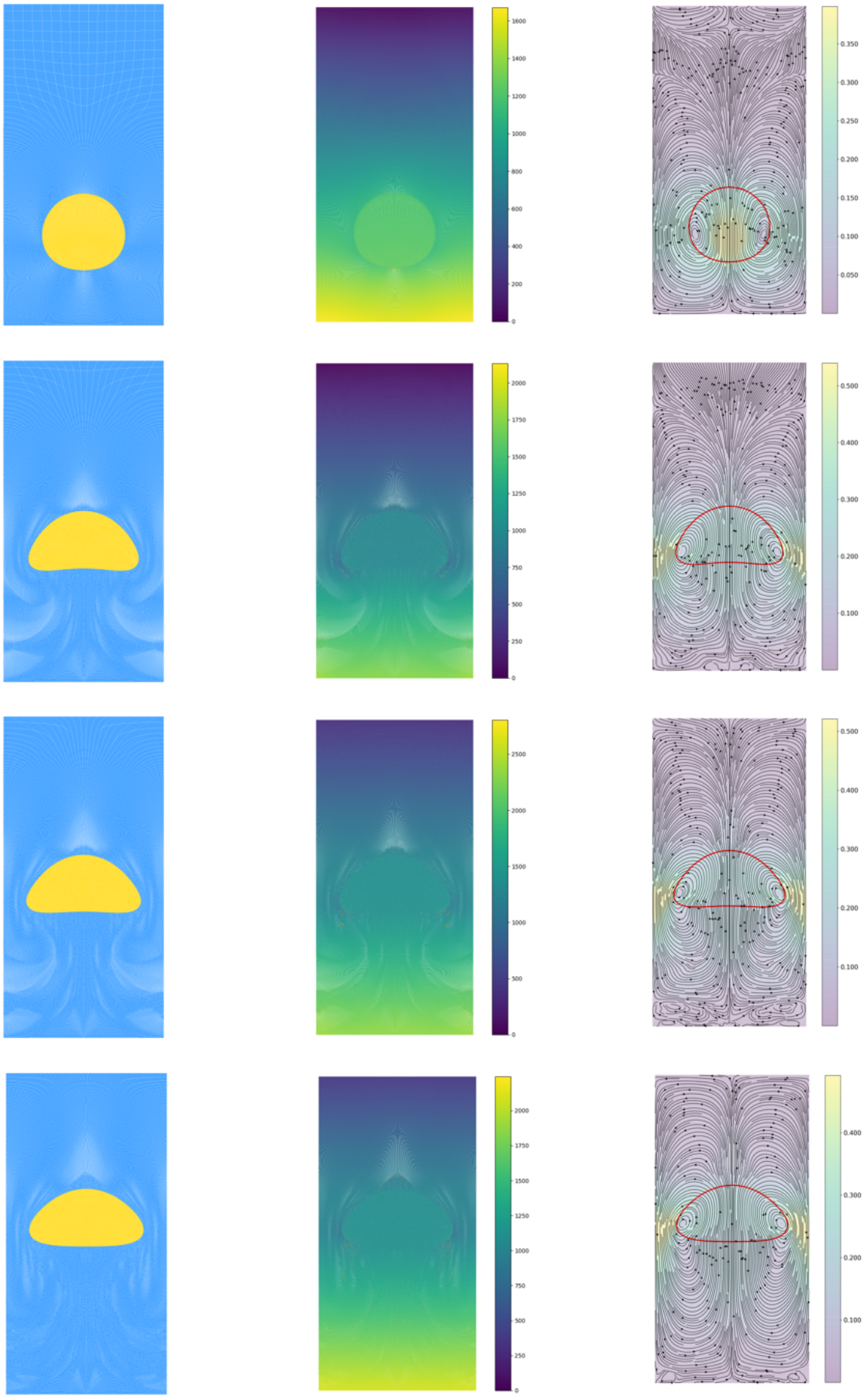
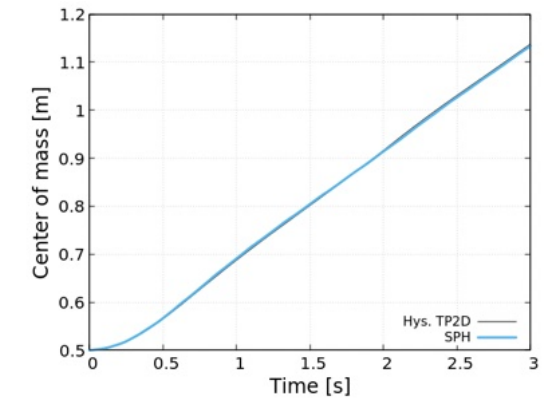


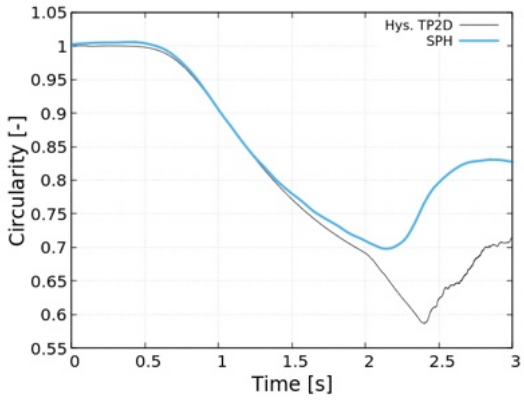
Figure 3.8: Ellipsoidal regime. From left to right: particle distribution, pressure field, velocity field and streamlines. From top to bottom, time instants: 0.6 s, 1.8 s, 2.2 s and 3.0 s.

3.3.2 Second case: skirted bubble

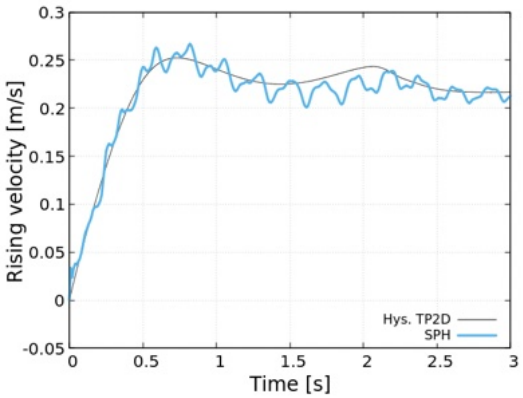
The second case is distinguished by domain attributes that give a skirted bubbly regime, parameters in Tab. 3.1. The density and viscosity ratios are much higher with respect to the 1^{st} case, we have $\rho_0^I/\rho_0^{II} = 1000$ and $\mu_0^I/\mu_0^{II} = 100$, which implies that this test is more challenging to reproduce.



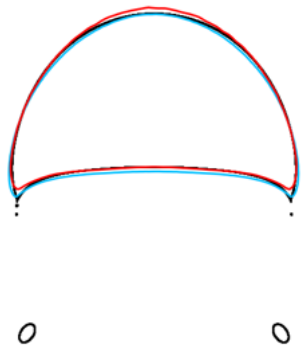
(a) Center of mass



(b) Circularity



(c) Rising velocity



(d) Shape at time 3s

Figure 3.9: Skirted regime, verification of results: [91] are denoted by the label Hys.TP2D (Transport Phenomena in 2D) and represented by a solid black line; SPH results are represented by solid light blue line.

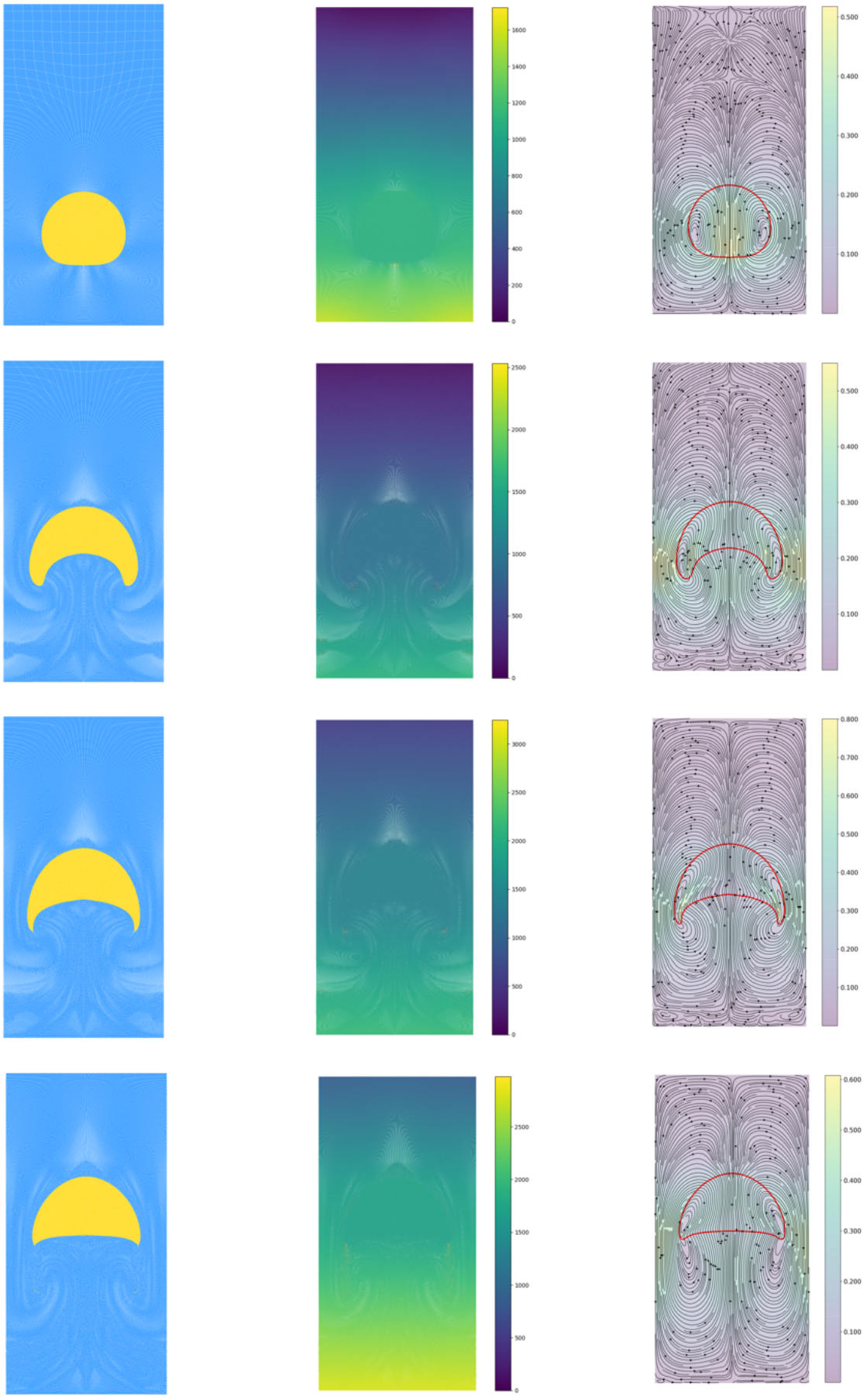
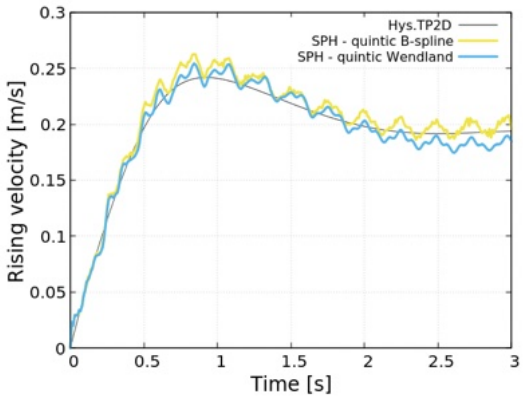


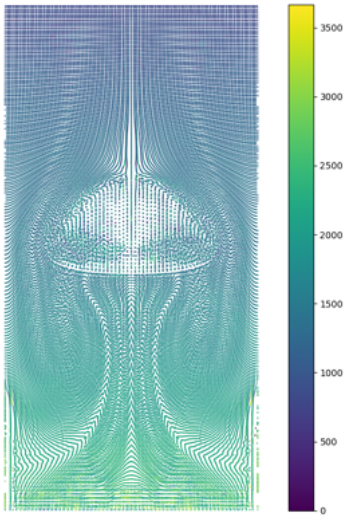
Figure 3.10: Skirted regime. From left to right: particle distribution, pressure field, velocity field and streamlines. From top to bottom, time instants: 0.6 s, 1.8 s, 2.2 s and 3.0 s.

3.3.3 Discussion

There are two pivotal aspects needed for detailed discussion in relation to the obtained results: the first one is the occurrence of oscillations in the rising velocity concerning both test cases; the second is solely pertained to the second test, and is the lack of an elongated skirt. Considering the extensive exploration of the benchmark by previous researchers, we have the opportunity to draw comparisons with the existing literature.



(a) Oscillatory rising velocities of interpolations given by quintic spline and quintic Wendland kernels: trend results are comparable.



(b) Pressure plot scatter at $t = 3\,\text{s}$, particle gathering issue is considerably visible.

Figure 3.11: First case by [91]: (a) comparison of oscillatory behaviour of rising velocity between quintic spline and Wendland kernels (b) pressure plot scatter at time $t = 3\,\text{s}$.

In Fig. 3.7c and 3.9c, the SPH behaviour is oscillatory, which is not surprising knowing that our approach allows for weak compressibility of the fluid: in weakly compressible approaches the pressure is determined by evaluating the variance in density which, despite changing very little during the simulation, can cause visible oscillations in the pressure nevertheless. We see oscillatory trend in the rising velocity in other SPH works, such as in Grenier et al. [77], in Zhang et al. [202], in Sun et al. [179], in Krimi et al. [97], in Yan et al. [199], in Arai et al. [5], in Patiño-Nariño [147] and very mild fluctuations also in the weakly compressible diffuse-interface method (Eulerian) by Kajzer [92] in his Ph.D. manuscript. The idea is justified noticing the complete lack of oscillations in truly incompressible approaches, Eulerian and Lagrangian, e.g., in the benchmark [91], but it is also corroborated in particle approaches as the SPH research conducted in 2018 by Rezavand et al. [162] and the work on MPS by Duan et al. [53]. Another factor that could account for the oscillations is the kernel: it is possible that the kernel itself may contribute to inaccurate results. Thus, we examined the first case by [91] using interpolation calculated by the quintic spline. Comparing the rising velocities obtained with both spline and Wendland kernels, as depicted in Fig. 3.11a, we observe very slight differences in the trends,

and they are possibly comparable. In conclusion, complete avoidance of this specific behaviour in the results is unattainable; however, mitigation is feasible, but it would necessitate further research into the influence of the kernel. Our implementation of spline functions is not conclusive, as shown in Fig. 3.11b, where the clustering of particles indicates an error that requires further investigation to rectify.

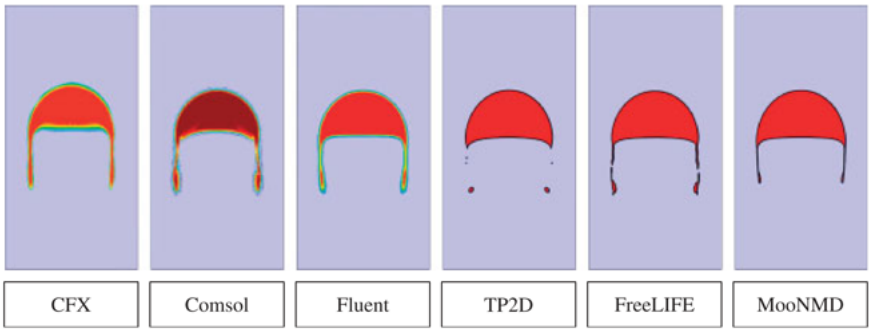


Figure 3.12: Set of snapshots (from [91]) at unit time $t = 3.0\,\mathrm{s}$ of the skirted bubble case implemented in six different software.

In our SPH bubble we can hardly notice the presence of the skirt, as it happens in the majority of works on the skirted regime implementing particle approaches, e.g. the paper on MPS [53] and mixed approach MPS-SPH proposed by Duan et al. [52] in 2020; also publications on SPH such as [179], [199] and [5]. The skirt formation does not appear to be a problem for truly incompressible SPH, see [162]. Besides, even in Fig. 3.12 taken from the benchmark, there is no ultimate behaviour of the skirt for which the entire community agrees; it highly depends on the type of method and algorithm used for the implementation. The complexity of the problem underscores the necessity for further in-depth investigations.

Moon pool 4

In this chapter we aim to demonstrate the effectiveness of the mVER formulation in tackling more demanding test scenarios. We attempt to model the moon pool, a well-known application in ships design, where the understanding of phenomena such as sloshing is fundamental. We initially tested our code against a sloshing tank benchmark and later implemented a simplified version of a moon pool.

4.1 Sloshing tank

In this section we present the implementation of sloshing in our SPH code. Sloshing is a term used to describe the movement of liquid featuring a free surface (or interface) within a tank when it is subjected to external forces like acceleration or deceleration. In numerics, understanding sloshing is crucial in various engineering and design applications, especially in structures that hold liquids like tanks, ships, or spacecraft. Precise modelling of sloshing allows engineers to anticipate and address potential issues, ensuring the stability and structural integrity of the container or vessel.

4.1.1 Literature review

Stepping into the extensive literature about wavy flow, there are some researches whose contribution has been crucial to the evolving understanding how implement sloshing within WCSPH method. The treatment of viscous force in existing literature often leans towards a simplified version, assuming the flow inviscid. Normally, dissipation is managed through the application of artificial viscous forces, a pattern affirmed by the majority of references in the subsequent list. Our objective is to showcase the robustness of our formulation despite the complexities posed by the challenging sloshing benchmark.

In the 2000s, researchers predominantly concentrated on replicating free surface behaviour and comparing results with Eulerian approaches.

The early exploration of sloshing, conducted in 2003 by Landrini et al. [100], laid the groundwork for subsequent research. This pioneering work tested sloshing induced by roll and sway excitations, the verification of SPH results showed remarkable agreement with the experimental data. The authors emphasised the need for future investigation on the SPH method's capability to handle fluid slamming against walls boundary. In the upgraded study of 2007 by Landrini et al. [101], the focus extended to splashing and near-shore bore propagation. The case simulated is a

flow with a semi-infinite layer of fluid, the wave is created by a vertical piston motion, the fluid run-up an inclined beach where the breaking bore phenomenon and resultant vortexes occur. A comparison between a free-surface implementation of SPH and the Boundary Element Method (BEM) solver is showed. Although the BEM solution had a less irregular behaviour compared to SPH results, probably thanks the the re-gridding process, the two methods emerged to be comparable, underscoring the robustness of the SPH approach. In both studies, the ghost particles method for boundaries is implemented, it is particularly useful to achieve an exact free-slip condition on the walls.

In the work by Colagrossi et al. [38] published in 2008, the focus lies in examining the dependence on the Reynolds number for shallow depth sloshing flows within rectangular tanks under forced harmonic motion. The initial simulation involved three substances: water, sunflower oil and glycerine. However, to prevent instabilities the water case would have consisted of millions particles, which is not feasible even in our days. Thus, only the second and third fluids are analysed. The results are later compared with Navier-Stokes level-set solver. The year after, a new publication by Colagrossi et al. [34] explored what were the free surface formulations that influence the smoothed differential operators in SPH. Furthermore, the principle of virtual works is employed to provide a framework to establish a clear link with the enforcement of dynamic free-surface boundary conditions. The free surface exhibits a commendable agreement with results obtained from a BEM. The study of 2010 by Colagrossi et al. [36] delved into the specifics of violent sloshing wave impacts, employing an enhanced SPH method, with Moving Least Squares (MLS) integral interpolants, that further refined the model. The flip-through phenomenon, a steep wave encounters a vertical wall, was observed in detail. In the same year, the work by Antuono et al. [4] focused on free-surface flows introducing an updated formulation of the diffusive term in the continuity equation, the so-called $\delta-$SPH firstly presented by Molteni and Colagrossi [122], whose aim is to mitigate spurious high-frequency oscillations within the pressure field. The term $\delta-$SPH is the approximation of the second-order derivatives, meant to enhance the accuracy of simulations. Regrettably, the formulation proposed in [122] has demonstrated inconsistency for particles near the free surface; its inconsistency manifested as an outward force on the fluid, leading to an expansion in volume and subsequent displacement of particles away from the phase. The application of these advancements extended to the study of the flip-through effect in sloshing, which is characterized by a sudden pressure increase at a focal point.

During the mid-2010s, there was noteworthy progress in addressing open boundaries, aiming to achieve a more realistic representation of sloshing phenomena. The 2018 paper by Tafuni et al. [187] the novel methodology for open boundary conditions in SPH is presented and integrated into the open-source code DualSPHysics. Regular waves are

initially generated using a piston-type wave maker and then utilizing the newly developed open boundary formulation. The 2019 study by Verbrugghe et al. [191] carried on the work done in [187] with the aim of simulating the wave flow in a reduced domain; achieving this goal involved implementing advanced boundary conditions capable of both active wave generation and absorption.

Recent investigations in the subject have chosen to incorporate the isothermal EoS rather than the Tait formulation, a trend observed even in the subsequent three references. We see this in the 2020 study by Krimi et al. [96], which introduces a WCSPH with the development of a modified particle shifting strategy (firstly proposed by [196]) able to mitigate the impact of truncated kernel support, eliminate tensile instability and prevent unwanted fluid fragmentation, mostly on the free-surface. To overcome the problem of voids or unrealistic phase separation when simulating high density ratio in WCSPH, Rezavand et al. [163] in 2020 proposed a multiphase method based on a low-dissipation Riemann solver and including the transport-velocity formulation in the lighter phase. Through the two-phase Riemann solver, the light phase experiences a heavy phase-like moving wall boundary, while the heavy phase exhibits a free-surface-like flow with variable free-surface pressure. The interesting outcome shows more stable interfaces and the lack of non-physical voids or interface separation. Comparative analyses with results obtained from BEM validate the effectiveness of the proposed method. The paper by Zheng et al. [203] published about two years ago, introduces an advanced WCSPH where the $\delta-$SPH term has been modified to handle both single- and multiphase flows. The proposed model is particularly well-suited for addressing long-term simulations, sloshing benchmark, for example. Results with and without the density diffusive term are compared, bringing out the importance of the correction: its exclusion can result in the accumulation of numerical errors over time, ultimately leading to the development of a non-physically oscillating pressure field in the later stages of simulations.

The papers [163] and [203] are particularly noteworthy for our purposes as they have successfully implemented multiphase SPH. In all the cited publications, the dissipation was managed using artificial viscous forces, except for [203], where the viscous term is discretised following the approach of [87].

4.1.2 The benchmark

Among the studies we have examined, the one by Green and Peiró stands out as particularly valuable, especially considering its thorough exposition in both the Ph.D. thesis [74] and subsequent paper [75].

In the work [75], WCSPH method is validated for simulating long-term sloshing flows in various geometry, and it is coded in the open-source

parallel code DualSPHysics. Three main issues have been investigated: the treatment of solid boundaries to minimize the introduction of numerical errors into the system; the necessity for a massively parallel implementation when simulating high resolution domain and the application of an accurate numerical dissipation scheme to mitigate existing numerical errors.

The fixed ghost particle technique by Adami et al. [1] has been employed to deal with solid wall boundaries. However, two minor drawbacks are associated with this method. When a particle approaches the wall, it is expected to be pushed away by the boundary force, but with this technique, the wall generates an attraction force that gradually pulls the fluid particle towards it. A possible solution to this issue involves establishing a thin buffer zone adjacent to the wall. The second concern revolves around the interpolation of pressure values for the fixed ghost particles. These pressure values, assigned based on a force balance on both sides of the solid wall, may encounter challenges when the kernel support is truncated, especially at the free surface. In such cases, the re-normalization of the kernel might not suffice to fully alleviate the error stemming from the approximation. Despite these two sources of inaccuracy, the method works nicely, particularly for complex geometry.

In the reference [75], the authors explore two dissipation sources in SPH: the viscous force in the momentum equation and the diffusive term added in the continuity equation and called $\delta-$SPH. In the paper, the viscous forces tested are the artificial viscosity coupled with free-slip boundary conditions and the laminar viscous force formulation (by Lo and Shao [197]). The implementation of the $\delta-$SPH scheme excludes the need for the artificial viscosity term, and it is used in the case of inviscid fluid flows. All of these dissipative terms serve the dual purpose of mimicking the physical dissipation and also act as an artificial damper to undesirable numerical oscillation in density and, therefore, in pressure. The interesting findings reveal that the $\delta-$SPH scheme effectively diminishes the numerical instabilities better than the artificial viscosity formulation.

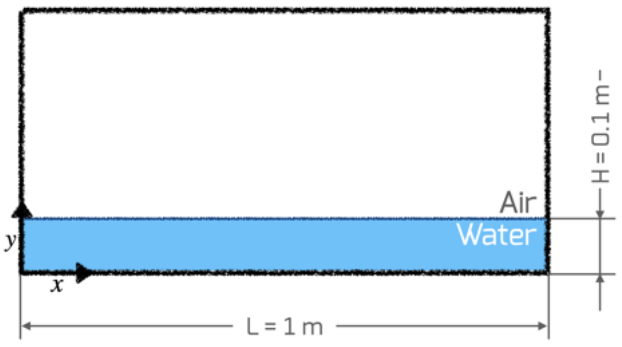


Figure 4.1: Sloshing domain from benchmark [75].

To validate our code for sloshing, we decided to reproduce the case by [75], but instead of running a free-surface (only-water) simulation, we

arranged the case to be multiphase. The fluid domain has a width of 1 m and a height of 0.5 m, as depicted in Fig. 4.1. Within this domain, a fine layer of water particles, measuring 0.1 m in thickness, is situated beneath a layer of air particles that is four times thicker. The water density and viscosity are $\rho_0^I = 1000\,\mathrm{kg/m^3}$ and $\mu_0^I = 0.001137\,\mathrm{Pa{\cdot}s}$, the air properties are $\rho_0^{II} = 1\,\mathrm{kg/m^3}$ and $\mu_0^{II} = 0.0000178\,\mathrm{Pa{\cdot}s}$. The surface tension coefficient is $\sigma_{st} = 0.0728\,\mathrm{N/m}$; for the sake of stability, the water artificial speed of sound is $s_0^I = 15\,\mathrm{m/s}$.

On the liquid phase acts a rectilinear sinusoidal force

$$\mathrm{f}_{\mathrm{b},x}(t) = a_0(t) = A(2\pi f_w)^2 \sin(2\pi f_w t) \tag{4.1}$$

where the displacement amplitude is $A = 0.5\,\mathrm{m}$, the excitation frequency is $f_w = 0.3 f_0 = 0.1462\,\mathrm{Hz}$ with $f_{(\mathrm{m}-1)} = \omega_m/(2\pi)$. Considering the first mode $\mathrm{m} = 1$, the natural frequencies f_0 depend on

$$\omega_{\mathrm{m}} = \sqrt{g\frac{\pi \mathrm{m}}{L}\tanh\left(\frac{\pi \mathrm{m}}{L}H\right)} \tag{4.2}$$

where g is the acceleration of gravity, L is the length of the tank and H is the water level. Given the wave characteristics, this scenario is akin to the case studied by [203], as the Reynolds numbers are of the same order of magnitude, specifically 10^5.

In order to determine the optimal resolution for the simulation, we compared the outcomes of two resolutions:

1. the lower resolution has smoothing length $h = 0.0078125\,\mathrm{m}$, the total number of particles is 32768, the number of particles composing the water phase is 6656;
2. the higher resolution has smoothing length $h = 0.0039062\,\mathrm{m}$, the total number of particles is 131072, the number of particles composing the bubble phase is 26112.

The results obtained from the simulation have been cross-checked with the analytical solution implemented for potential flow by [61], that gives for the wave height $\zeta(x,t)$ and the sloshing force $\mathrm{f}_x(t)$ the following formulations:

$$\zeta_s = \sum_{\mathrm{m}=1}^{\infty} \beta_{\mathrm{m}}(t)\cos\left(\frac{\mathrm{m}\pi}{L}x\right) \tag{4.3}$$

$$\mathrm{f}_{x,s}(t) = \rho H L a_0(t) + \rho\frac{L^2}{\pi^2}\sum_{\mathrm{m}=1}^{\infty}\ddot{\beta}_{\mathrm{m}}(t)\frac{1+(-1)^{\mathrm{m}+1}}{\mathrm{m}^2}. \tag{4.4}$$

The value of $\beta_{\mathrm{m}}(t)$ and its second derivative $\ddot{\beta}_{\mathrm{m}}(t)$ come from solving the following ordinary differential equation using the fourth-order Runge-Kutta scheme, and considering the modes of $\mathrm{m} = 1, \ldots, 25$

$$\ddot{\beta}_{\mathrm{m}}(t) + \omega_{\mathrm{m}}\beta_{\mathrm{m}}(t) = -a_0\left(\frac{2}{\pi \mathrm{m}}\right)\tanh\left(\frac{\pi \mathrm{m}}{L}H\right)[(-1)^{\mathrm{m}} - 1]. \tag{4.5}$$

Upon evaluating the outcomes of both resolutions against the analytic solution, the root mean square error in wave height comparison is 0.244% for the lower resolution and 0.219% for the higher resolution.

4.1.3 Results and discussion

The analysis focuses on a 30-second simulation, and the outcomes of the sloshing effect on the left wall are presented. The top plot in Fig.4.2 represents the wave height whereas the bottom plot stands for sloshing force, the SPH results are compared with analytic data obtained for potential flow in Eq. 4.3. The favourable agreement regarding wave height is highly promising; the sloshing force has a similar trend to the analytic, but the amplitude is underestimated due to the presence of the viscous force.

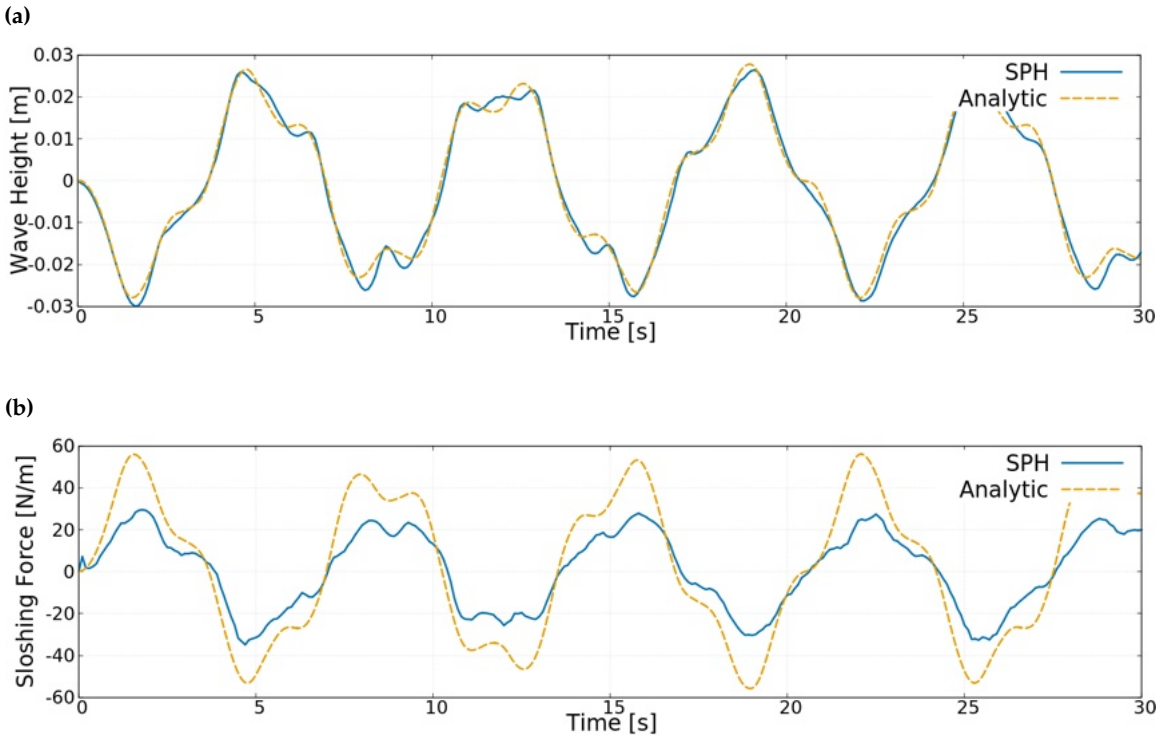
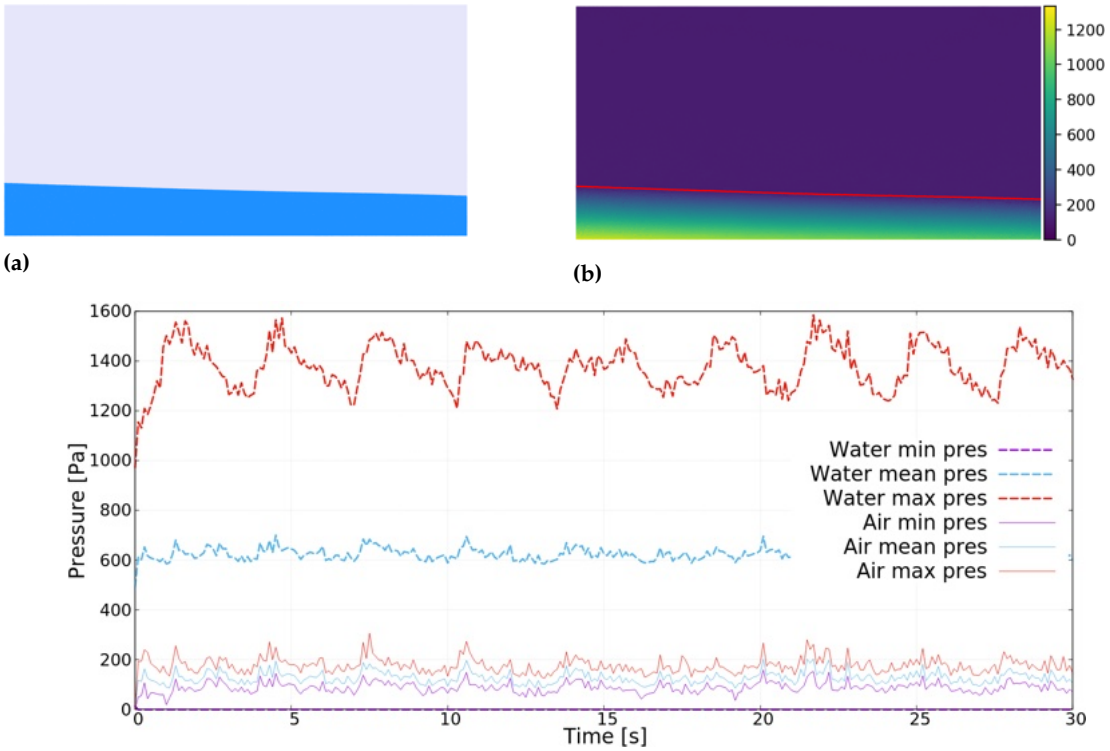


Figure 4.2: Sloshing tank benchmark, comparison between SPH results in blue and analytic formulation [61] in yellow: (a) wave height on the left wall of the tank, (b) sloshing force on the left wall of the tank.

In Fig. 4.3, a snapshot at a later time is presented: On the left, the distribution of particles at 18 s, and on the right, the corresponding pressure field. The particles exhibit a uniform distribution, and there are no apparent instabilities in the pressure field. This trend is affirmed in Fig. 4.3c, where only slight oscillations are evident, attributed to compressibility effects. The pressure within the air phase remains stable and of relatively low magnitude, underscoring its minimal impact on the overall simulation.



(c) Pressure trend in water and air phase.

Figure 4.3: (a) Particle distribution at time 18 s, in (b) the corresponding pressure field. Below, (c) represents the pressure distribution in time: minimum, mean and maximum value for both water and air phases.

4.2 The moon pool

In agreement with ESTECO SpA, the company partner of the present research, we started working on the moon pool case.

The moon pool is an opening through the hull of vessels, it allows underwater activity from on-board. It is typical of drill ships (e.g., Samsung Heavy Industries SHI ships), offshore platforms, semisubmersible vessels, research vessels (e.g., the RRS *Sir David Attenborough*, the RSV *Nuyina*, etc.), barges and other. Maritime studies often focus on surface wave phenomena, interaction between solid body and fluid, rotational and translational motion of the ship. Experiments on ship dynamics are quite complicated and usually require the design and manufacturing of a scaled model. For this reason, the CFD is widely used in marine engineering research.

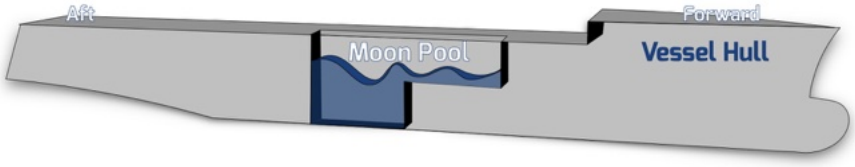


Figure 4.4: Schematic representation of a ship encompassing the moon pool in the center of the hull (inspired by [81]).

In this section, we present the case delving into its historical context;

we partly analyse the literature concerning the subject to define which study to consider as a benchmark. Finally, we describe our SPH model in details and present the corresponding outcomes.

4.2.1 Background

The evolution of the moon pool concept is closely tied to progress in offshore technology and the need for efficient underwater exploration and operations in drilling industries.

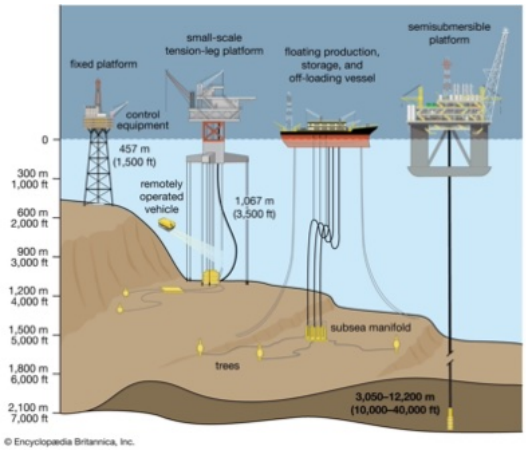
The history of drilling began in the mid-19th century in the oil sector, the first onshore well in Pennsylvania (USA), is credited to Edwin Laurentine Drake and dates back in 1859. In 1987 in California, a pier supporting a land rig was build in California, marking the start for the offshore drilling. The first offshore well was drilled in the Gulf of Mexico in 1947, the design of submersible of the time allowed for a maximum depth of only six meters of water[1].

1: For reference, Offshore online magazine [139].

The first contemporary drill ship is the CUSS 1 inaugurated in 1961 and named after the collaboration of Continental, Union, Superior, and Shell oil companies that developed it in 1956[2]. In the ship were included four rotating thrusters, one at each corner, and it was the equipped with the dynamic positioning, a computer-controlled system to automatically maintain a vessel's position and heading by using its own propellers and thrusters. The dynamic positioning utilized submerged sonar buoys arranged in a circular pattern around the vessel, receiving signals to sustain a footprint of about 600 feet (about 180 m) in diameter. The deployment of equipment into the water was enabled by the moon pool for the first time. The moon pool became integral to drill ship design, enhancing stability and creating a safe environment for underwater drilling operations.

2: For reference, Dynamic Positioning Committee [56].

(a) Picture of a moon pool



(b) Offshore drilling platforms

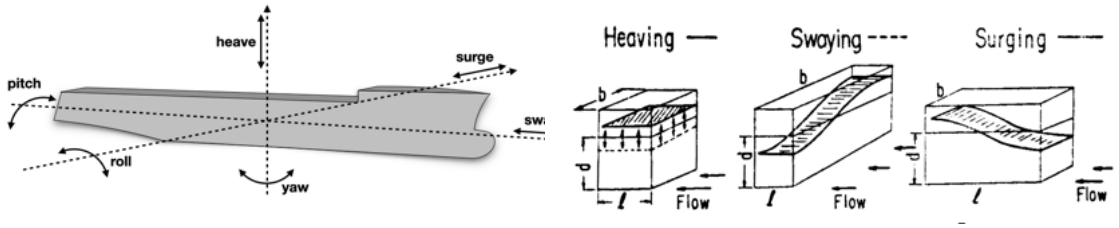
Figure 4.5: (a) Picture of a moon pool in a drill ship, from BOURBON [17]. (b) Offshore drilling platforms from Britannica [57]: sampling of technologies used to recover petroleum from the sea.

There are fixed rigs and floating rigs in modern industry, in Fig. 4.5b we see a sample of them. Bottom-resting platforms are typically utilized for well drilling in fields or in waters shallower than 914 m; conversely, floating rigs can exceed this limit. Fixed platforms, which rest on the sea floor, are very stable but are limited in their ability to drill in deeper waters compared to floating platforms. The most common type of fixed platform is the jack-up rig; these platforms do not have propellers, and they are fastened to the sea floor with legs[3]. In the early years, all rig types operated with support from the seabed during drilling and included inland barges (to 3 m), drilling tenders (to 12 m), submersibles (25 to 30 m), drill barges (to 35 m) and self-elevating, or jack-up units (to 45 m). Platforms designed for deep-water production encompass tension-leg systems and floating production systems (operating beyond depths of 1830 to 2130 m), capable of vertical movement in response to ocean conditions. Semisubmersible platforms, suitable for ultra-deep waters ranging from 3050 to 12200 m, float thanks to an entirely submerged hull. The operational platform remains well above the water's surface, minimizing the impact of normal wave action. These platforms are typically secured in position during drilling by cables fastened to the sea floor. The position of a platform is usually decided on the sea bed analysis done with the drill ships, which are ships specifically designed for the exploration. These ships are exposed to roll and pitch from wave action, which makes the drilling difficult. Typically, these vessels are mounted with a derrick for the drilling operation in the middle over an opening, meaning the moon pool.

3: For reference, Britannica Encyclopaedia [57].

The vessels at sea have six degrees of freedom, three of them are translational and the remaining three are rotational, we see a schematic representation in Fig.4.6a. Shortly, the translations are as follows: the surge is the forward and backward motion, potentially stemming from the propulsion system; the sway is the sideways motion, often induced to water and wind acting perpendicular to the hull; and the heave is simply the upward and downward motion caused by ocean waves. The rotations are indicated as follows: rolling occurs when the vessel turns about its longitudinal axis due to manoeuvres or to wind pressure on sails; pitching involves rotation around the transverse axis, and yawing around the vertical axis. Examining all the degrees of freedom of a vessel simultaneously through numerical and experimental analyses can pose challenges due to the substantial number of variables and their interdependencies. Moreover, some of these effects are significantly reduced when the ships are stationary, as is the case with drill ships during drilling operations. Therefore, employing a simplified model is helpful.

The moon pool has several advantages: firstly, it offers a secure environment during drilling operations, facilitating the deployment of equipment at sea; secondly, its central location prevents the instability that would otherwise occur if operations were conducted from the side



(a) Degree of freedom of vessels

(b) Water motion inside the well

Figure 4.6: (a) Schematic representation of the Degree of freedom of vessels in open sea, (b) water motion inside the well, from [68], the heave corresponds to the piston motion, swaying and surging to sloshing.

of the ship. Unfortunately, the moon pool affects the stability of the vessel in other ways, both when it is sailing and when it is not moving but is subjected to sea waves. The water motion within the moon pool can be categorized in two ways: piston motion, the vertical rise of water mostly visible in the square-shaped pool; the sloshing motion, the horizontal oscillations of the water between the walls. In a rectangular-shaped pool these two effects are mixed. Besides these problems, in heavy sea often occurs the green water phenomenon, which is a large quantity of water over-topping the deck.

Depending on the application, the moon pool can either be at sea level, which is the case for drilling, or positioned below. Recently, the inclusion of a step known as a recess inside the moon pool has proven beneficial, particularly for assembling drilling equipment. Despite its utility, the recess can introduce unwanted effects, underscoring the need for careful scrutiny in its design. In general, the research on the topic reveals that crafting an effective and high-performing moon pool is a complex task, it is highly dependent on the sea conditions where the vessel is intended to operate.

4.2.2 Literature review about the moon pool

Likely, the first paper about the moon pool was written by Fukuda [68] and published in 1977. The author built four different model ships featuring vertical openings, both rectangular and circular, and carried out diverse experiments with and without constraints on the ship's sway motion. The study of the water surface inside the well revealed the occurrence of heave motion (currently called the piston motion) and showed that the water regularly moves even in the other directions, according to the surge and sway phenomena (resulting in sloshing). Moreover, the research compared the experimental findings with analytical analysis, demonstrating agreement on the calculation of the heave amplitude and on the increase of the drag caused by the presence of the well itself. The research highlighted a significant factor: the fluid dynamics within the well is influenced by both its shape and the characteristics of the stream. Lastly, the author suggested the employment of short flanges on the vertical walls of the well to mitigate the water motion.

In 2001, Molin [121] utilized linearised potential flow theory to investigate the water surface in a rectangular moon pool of barges with infinite length and beam, considering two- and three-dimensional scenarios. The outcomes revealed that the frequency of the sloshing modes are dependent on the width and the draught of the moon pool.

In 2007, Faltinsen et al. [60] presented a work on the potential flow theory that was once again applied to study two-dimensional piston-like steady-state motions of a fluid within a moon pool formed by two rectangular hulls. To avoid mathematical complexities that could arise from complex-shaped moon pools, the authors focus on a simpler scenario, namely a two-dimensional rectangular-shaped moon pool undergoing heave motions with a small amplitude (calm water). The comparison between the theoretical predictions and experimental results showed satisfactory agreement, attributing the differences to free-surface non-linearity.

The publication by Gaillarde and Cotteleer (2004) [70] involves employing CFD to address the fluid dynamics within the moon pool. They applied the Volume of Fluid (VOF) algorithm and solved the incompressible Navier-Stokes equations with free-surface conditions.

In the 2007 publication by Park et al. [145], numerical and experimental analyses on moon pool operating in rough seas are conducted. The main focus is on studying the consequences of sloshing leading to deck overflow. This occurrence is mitigated by the addition of blocks along the moon pool walls, which dampen the waves. An additional experimental study, complemented by numerical evaluation, is presented in the work by Son et al. (2008) [175]. This study focuses on the presence of a recess, a step incorporated into the moon pool. The recess serves multiple purposes, including facilitating the assembly of drilling equipment and potentially reducing the drag on the ship when moving forward. A noteworthy investigation was conducted by Huang et al. (2020) [89], where the authors experimentally examined methods to diminish the oscillatory movement in a moon pool with a large aspect ratio. In the study, the effectiveness of four damping devices is tested: a positive grid of flaps, a negative grid of flaps, a grid of baffles, and a vertical bulkhead.

The master thesis by Hammargren [81], held in 2012, is still one of the most important works on the subject. As pointed out, moon pool is meant to be a safe environment during operating time; however under certain conditions, e.g. the ship transit or heavy sea, the waves inside it may have greater amplitude than those outside, reproducing the effects of piston motion and sloshing. The presence of a moon pool results in an increment of drag force, consequently reducing the ship's speed and causing an up-tick in fuel consumption. Several devices are designed to reduce wave disturbance within the moon pool; some commonly used options are the creation of wedges under the moon pool, flaps, vertical bulkheads, flanges, damping chambers, movable structures, moon pools

with convergent shapes and moon pool with recess. The study involves the modelling of mainly two model vessels during transit conditions; they differ in the type of the moon pool, designed either with a simple rectangular shape or with a recess type. The findings validate that the recess has the capability to reduce drag.

4.2.3 The benchmark

Among the researches that we have reviewed, our choice leans towards the recent work conducted by Guo et al. [79], published in 2017, for the reasons explained in the following.

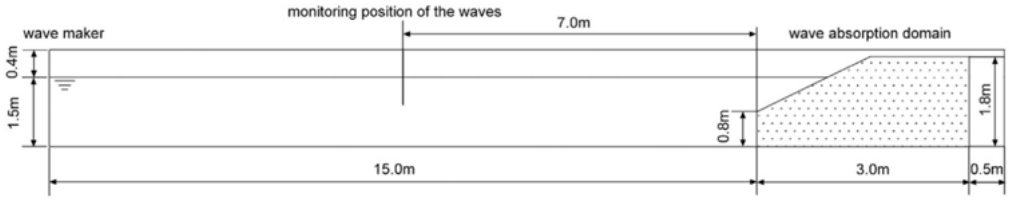
Given the relatively uncharted nature of the issue, having experimental data juxtaposed with numerical results remains crucial. This paper not only presents numerical findings but also includes experimental data gathered at the deep-water offshore basin at the Shangai Jiao Tong University (see Fig. 4.7c). The numerical configuration, in Fig. 4.7a comprises various details aimed at enhancing alignment with the experimental set-up: the flow is simulated utilizing the Airy wave theory in two dimensions; for the wave absorption domain, a Sommerfeld radiation outflow boundary condition is integrated with a porous media model to prevent wave reflection.

This work recognizes the current prevalence of deep-water drill ships featuring large rectangular moon pools rather than circular or small rectangular ones. It emphasizes also the importance of locating the moon pool in the midship position to minimize the consequences and influences of the the external flow. In addition, the moon pool features a recess which is advantageous in nowadays vessels for arranging drilling equipment.

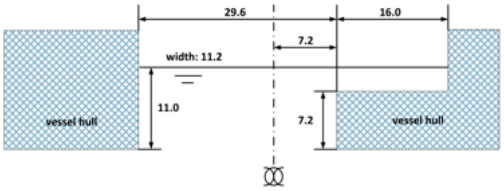
Moreover, in this research, two simplifications have been considered, proving beneficial in modelling the case within our SPH code. Both the experimental and numerical models are scaled at a ratio of 1:50 of a full-sized drill ship. The numerical vessel is stationary (i.e. moored in the experimental set-up), ensuring that the water behaviour within the moon pool remains minimally influenced by the natural heave or the pitching period resulting from the external flow.

Table 4.1: Main measurements of the full-scale and model (1:50) drilling vessel, from [79].

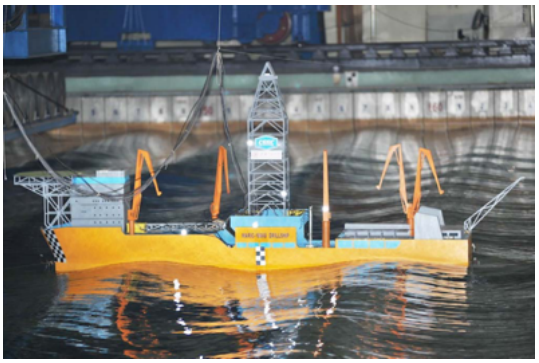
Parameter	Unit	Prototype	Model
Length between perpendiculars	m	180.80	3.616
Breadth	m	32.20	0.644
Depth	m	17.00	0.340
Draft	m	11.00	0.220
Displacement	ton	51,000	0.3981
Centre of gravity above keel	m	12.50	0.250
Centre of gravity forwards from midship	m	-0.757	-0.015
Transverse radius of gyration	m	11.49	0.230
Longitudinal radius of gyration	m	52.65	1.053



(a) Numerical basin implemented by reference study



(b) Full-scale moon pool



(c) Experimental set-up

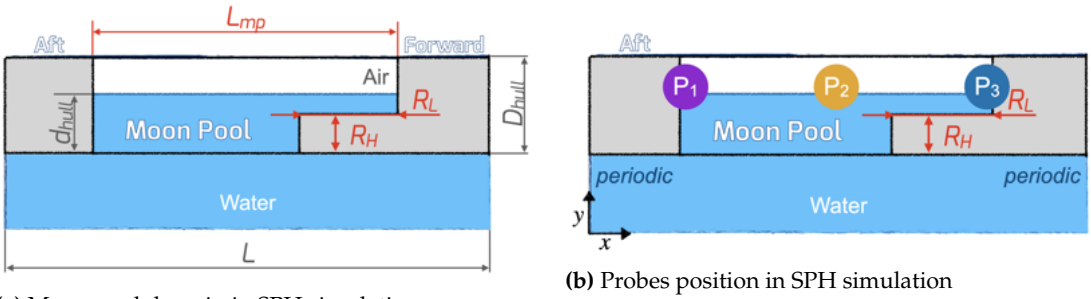
Figure 4.7: (a) Size of the numerical basin implemented by the benchmark study. (b) Full-scale moon pool from the benchmark, it must be scaled 1:50. (c) Picture of the experimental set-up. Taken from [79].

The three-dimensional numerical analysis was performed in the commercial computational fluid dynamics code Flow-3D, the interface was handled using the VOF technique. The incident wave frequency was calculated using the prediction equation by [121] for longitudinal sloshing motion, and it was equal to 87 Hz. This frequency is the resonant frequency of water motion within the moon pool based on its dimensions.

4.2.4 SPH model

Since SPH is particularly efficient for violent impact flows and macro-scale and multiphase problems, we think it might be a viable alternative to VOF in selected applications, including the moon pool case. In our early paper published in 2022 [177] we have presented a first attempt at the issue, wherein the modelled 2D moon pool is of the same dimensions as a full-scale one. Later, our interest in the work by [79] has led to the scaled configuration, comparable with the reference. However, SPH properties and the capabilities of our code are insufficient for an exact replication of the benchmark [79]; additional simplifications are necessary. We explain them in the following.

In SPH, conducting a simulation that considers a basin of the benchmark's size in three dimensions, with sufficient resolution, would result in a domain involving a substantial number of particles, which implies an exceptionally long simulation duration. Our SPH domain, schematically represented in Fig. 4.8, is two-dimensional, the size in the x-direction is $L = 2\,\mathrm{m}$ while in y-direction is $H = 1\,\mathrm{m}$; it is a close-up focused on the



(a) Moon pool domain in SPH simulation

(b) Probes position in SPH simulation

Figure 4.8: (a) Schematic representation of SPH domain. (b) Schematic representation of SPH boundary conditions and probes location (P_1, P_2, P_3) within the moon pool.

moon pool location, the moon pool bottom line is centred with the center of the domain in both directions. The dimensions of the hull and moon pool match those of the benchmark, the hull depth is $D_{hull} = 0.34\,\text{m}$, the draft is $d_{hull} = 0.22\,\text{m}$ (see Tab. 4.1), and the top line of the moon pool measures $L_{mp} = 0.912\,\text{m}$. The recess is characterised by height and length that respectively measures $R_H = 0.144\,\text{m}$ and $R_L = 0.32\,\text{m}$.

Our code lacks the implementation of open boundaries; consequently, in the simulation, we treat the side boundaries as periodic. The boundary condition on the top and bottom walls is set to free-slip, acknowledging that the basin depth surpasses the depth of our domain. The vessel is treated as captive, which in the simulation translates to a fixed hull: the particles composing the hull do not move, thus they can be considered as dummy particles.

The aforementioned constraints prevent the inclusion of a wave maker and wave absorption conditions in our SPH simulation. Hence, given the successful integration of the sloshing effect into the code, as demonstrated in Section 4.1, the external excitation in Eq. 4.1 involves a straightforward horizontal force that undergoes periodic changes in direction. Based on the wave properties of the reference, the displacement amplitude is $A = 0.04/2\,\text{m}$ and the excitation frequency f_w corresponds with the resonant frequency $0.87\,\text{Hz}$. As a result, what will be observable in the moon pool is predominantly the sloshing motion rather than a combination of sloshing and piston motion.

Since our primary focus is on simulating multiphase flow, we include both the air and water phases, with their properties matching those outlined in Section 4.1, except for the artificial speed of sound, which is in water phase equal to $35\,\text{m/s}$.

4.2.5 SPH results and discussion

In order to determine the optimal resolution for the moon pool case considered, we compare the outcomes of two simulations:

1. the lower resolution has smoothing length $h = 0.015625$ m and total number of particles equal to 32768, where 18922 water particles, 6992 air particles and 6854 hull particles;
2. the higher resolution has smoothing length $h = 0.0078125$ m and total number of particles equal to 131072, where 75606 water particles, 28246 air particles and 27220 hull particles.

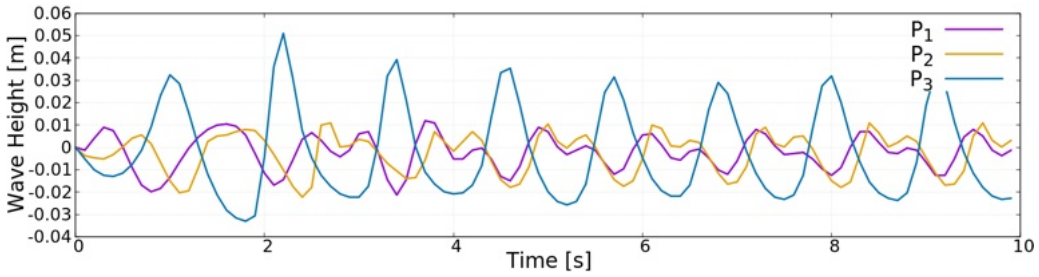


Figure 4.9: Wave height oscillation in time each probe.

To monitor the water level within the pool, we position three probes along the air-water interface as seen in Fig. 4.8b: probe P_1 is located at the aft side of the hull, the second probe P_2 is positioned at the center, and P_3 is located on the forward side of the hull. The sloshing phenomenon is evident in Fig.4.9: after approximately about 2 s transient, the flow reaches the steady-state regime. P_1 and P_3 have inverse crests and troughs, which is obvious because the wave travels in one direction and then moves in the opposite way. P_1 and P_2 show a more complex oscillation compared to the third probe, but the maximum water elevation is reached in P_3. The presence of the recess is likely the cause of this behaviour, confirming the conclusions in [79]: the recess acts as an artificial local sea bottom with finite water depth, thereby shallow water waves travel over the obstacle with sharp wave crests and shallow troughs.

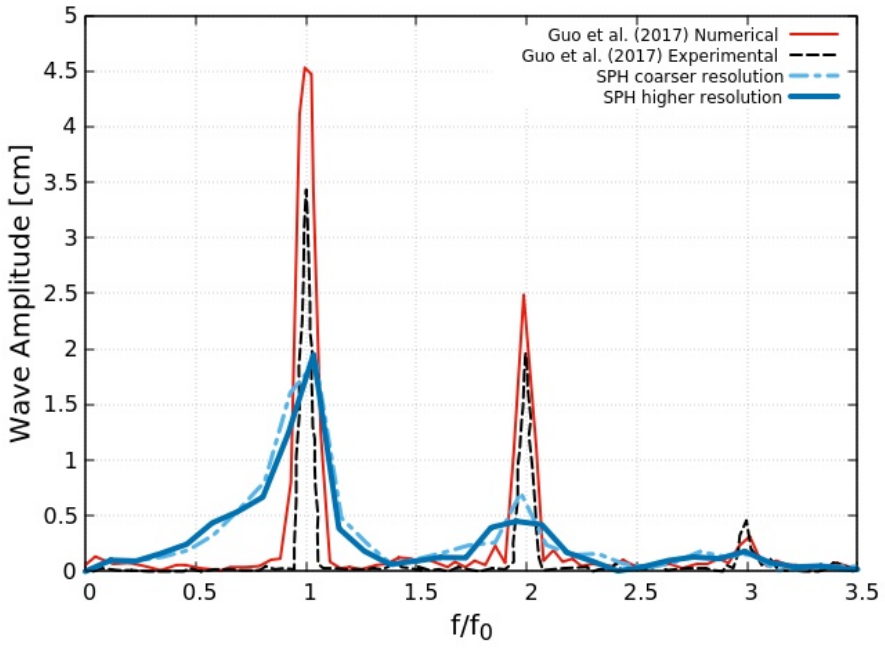


Figure 4.10: Dominant frequency comparison between the benchmark [79] and the SPH results, obtained with FFT analysis. On the x axis the dominant frequency over the reference value, the resonant frequency $f_0 = 0.87$ Hz; on the y axis the amplitude of the signal.

The SPH results are compared with the benchmark data, both numerical and experimental. Focusing on P_3, the probe that shows the most critical behaviour as a consequence of the recess presence, in Fig. 4.10 the implementation of the Fast Fourier Transform (FFT) verified that the dominant frequency corresponds to the resonant one: the dominant frequency obtain in SPH is equivalent to 0.90 Hz, which is about 3% higher than the reference $f_0 = 0.87$ Hz. The peak in SPH aligns with the reference, yet the amplitude is notably smaller. Reasonably, the difference in amplitude could be attributed to the simplifications taken into account: we considered a small two-dimensional domain subjected to linear excitation rather than a three-dimensional tank with a wave maker.

The sequence of pictures in Figures 4.11, 4.12 and 4.13 represents a set of snapshots of time instants between 6.0 s and 7.2 s in a quasi-periodic regime. The images on the left depict the distribution of particles for each phase, with the exclusion of solid particles from the hull. Meanwhile, the images on the right consist of scatter plots representing the corresponding pressure field, which appears to be consistently uniform. Examining Fig. 4.9, we observe that the oscillation period is slightly over one second, comparable to the sloshing frequency induced by the external force.

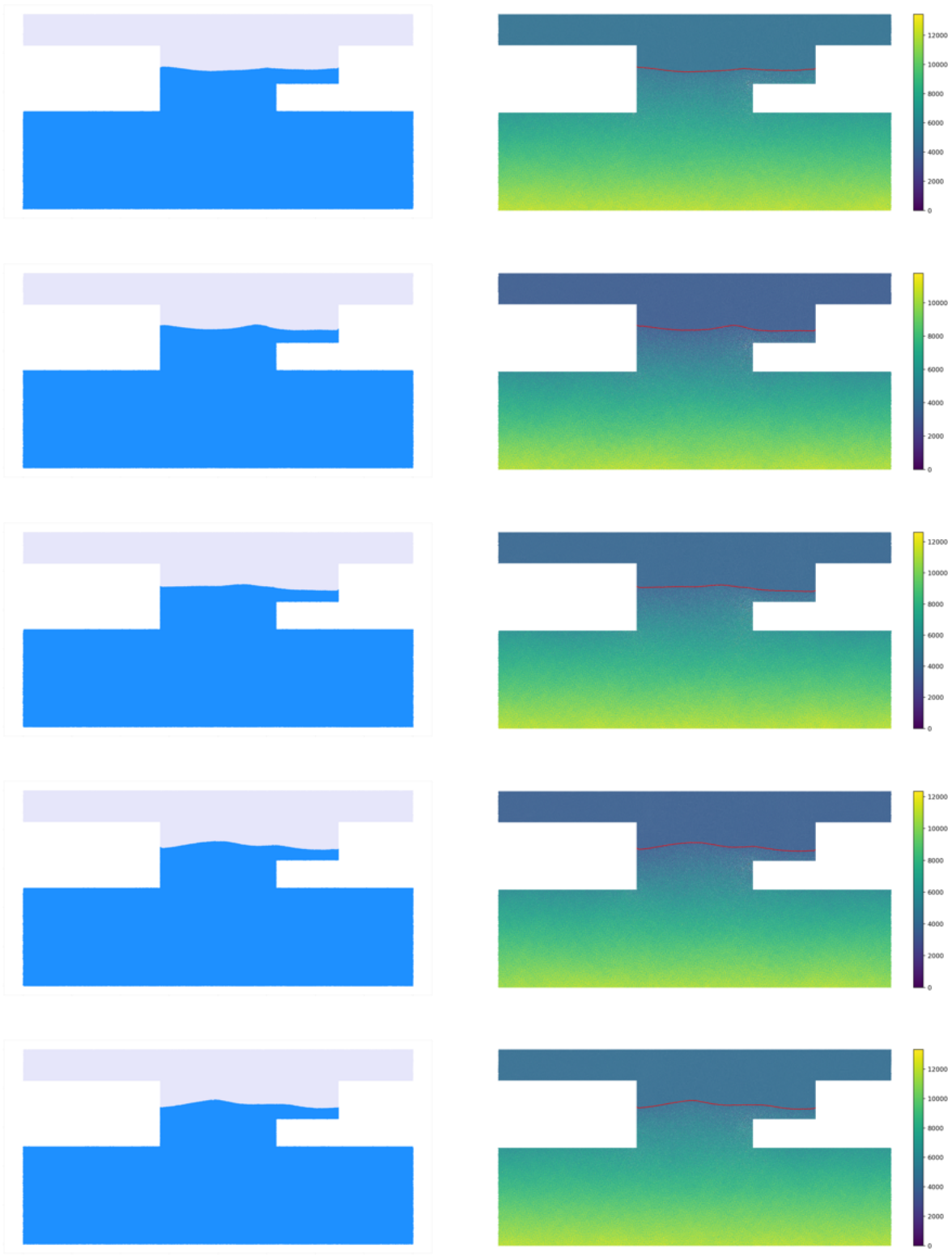


Figure 4.11: On the left side, particle distribution of water and air phases; on the right side the corresponding pressure field. Time instants: 6.0 s, 6.1 s, 6.2 s, 6.3 s and 6.4 s.

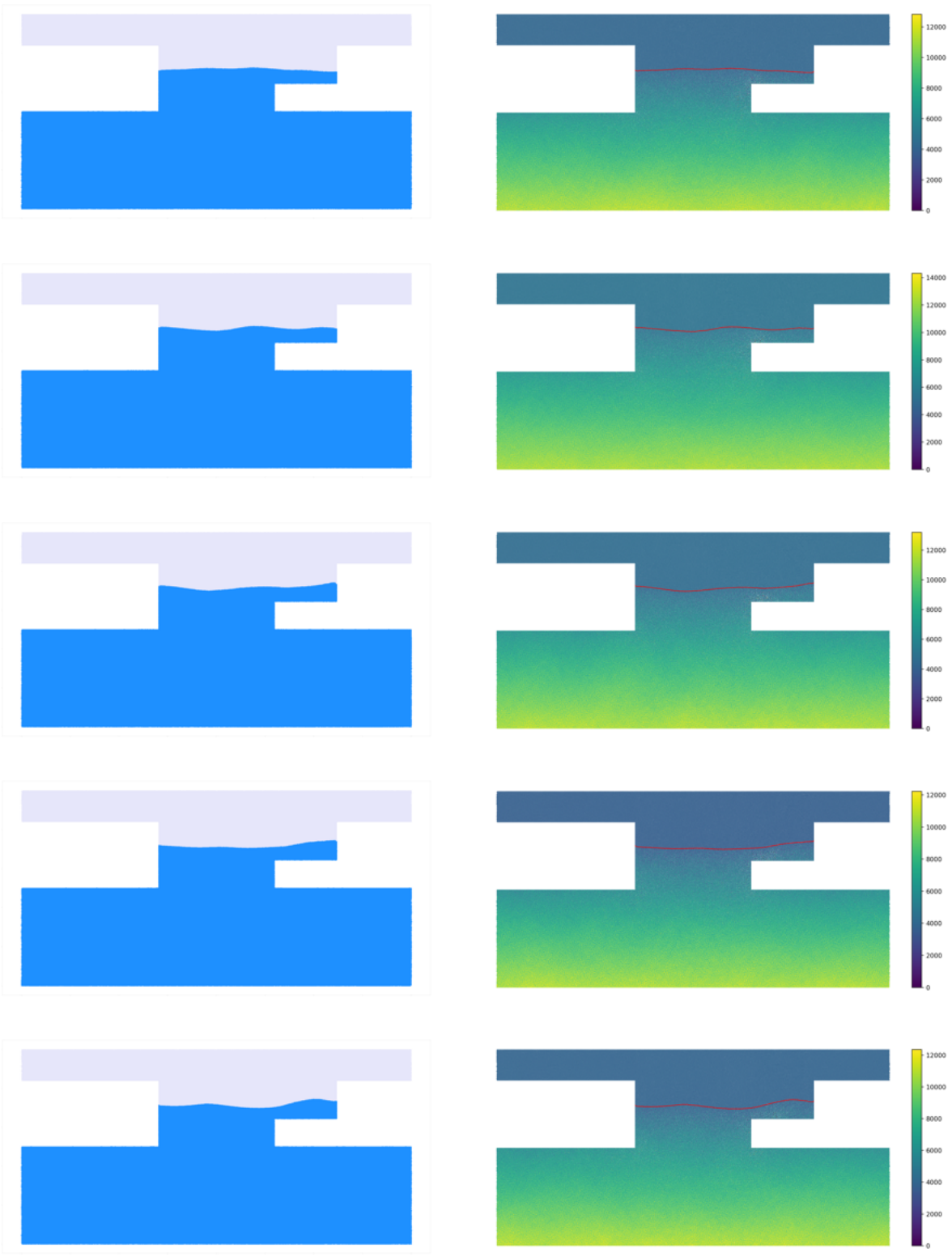


Figure 4.12: On the left side, particle distribution of water and air phases; on the right side the corresponding pressure field. Time instants: 6.4 s, 6.5 s, 6.6 s, 6.7 s, 6.8 s and 6.9 s.

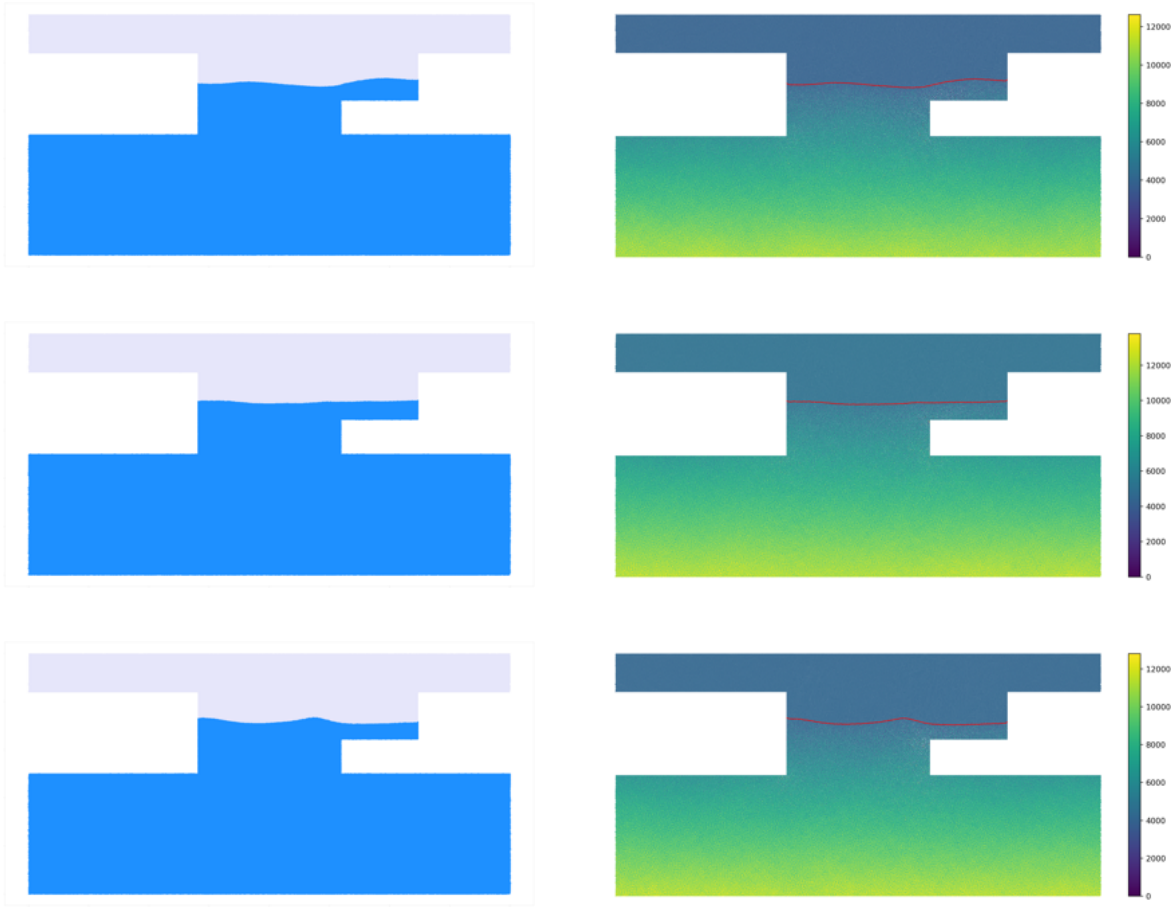


Figure 4.13: On the left side, particle distribution of water and air phases; on the right side the corresponding pressure field. Time instants: 7.0 s, 7.1 s and 7.2 s.

Optimization process applied to the moon pool case

5

In this chapter, we will discuss how to minimise the wave height inside the moon pool by optimising the shape of the recess with the software modeFRONTIER. We will cover the fundamental principles of mathematical optimization and their implementation in the software, explaining, step by step, how we used the latter for optimising the recess geometry.

5.1 Insights on mathematical optimization

Mathematical optimization is the procedure of seeking the optimal or most convenient solution to a mathematical problem. Citing Rockafellar in his lecture notes [164], the optimization process entails fine-tuning variables or parameters within a mathematical model represented by a function, which allows for comparison of the different choices and identifies the optimal one. Optimization is used in various fields, such as mathematics, engineering, economics, and computer science, to improve or optimize the performance of a system, design, or process.

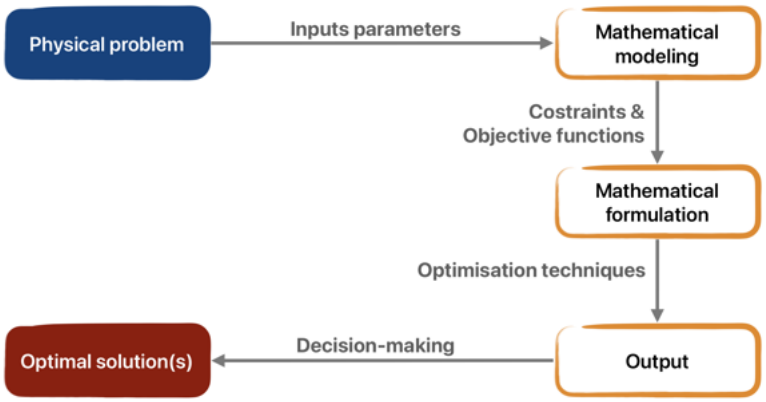


Figure 5.1: Basics steps of the optimization process (from [134]).

In Fig 5.1 we see the flow chart taken from the book by Nayak S. [134] (2020): the scheme represents the optimizing procedure, which starts with the analysis of the physical problem and is usually referred to the word *design*. The difficulty lies in determining the input parameters that have the most and least impact on the results.

The mathematical definition of the objective function is to be defined and maximised or minimised to determine the outputs. E.g. in industry, often the objective is to minimize the overall cost of the production and/or to maximize the profit [134]. In addition to the objectives, constraints can be incorporated into the model to establish the conditions or boundaries that ascertain whether a solution is acceptable or not. The optimization process develops an algorithm, the goal of which is to find a small set of

eligible and reasonable solutions. The ultimate selection among the set of solutions is usually subject to the type of application.

Although in Fig. 5.1, the optimization may seem straightforward, in fact, it is not. The dependence on the input is often hardly predictable, especially when the number of initial parameters is large. In addition, a good optimization problem needs an adequate number of iterations to be satisfied. For these reasons, it is useful to make the process automated through software.

The reliance on input is often challenging to predict, particularly when dealing with a large number of initial parameters, and moreover, it may require a tailored number of iterations to succeed. For these reasons, employing software to automate the process proves to be advantageous. We had the opportunity to use the software modeFRONTIER (mF), one of the flagship products of the Italian company ESTECO SpA [176]. ESTECO is a software provider in Trieste, and has twenty years of experience in the field of design optimization and data management. The company interacts regularly with the University of Trieste, which explains why its products fit academic and industrial needs.

5.1.1 A bit of history

The history of mathematical optimisation is a fascinating journey that starts in antiquity and spans various centuries and disciplines[1]. The mathematicians of ancient Greece were already testing optimising techniques mostly connected to their study in geometry; for example in the 300 b.C. Euclid demonstrated that a square has the largest area among rectangles of a given perimeter while he was studying the minimal distance between a point and a line.

1: For reference, Aalto University [90].

In the 17th and 18th centuries, the approach to optimization problems began incorporating methods from calculus. In this period, prominent figures such as Kepler, Galilei, Newton, Euler, Lagrange, and many others experimented with the need of optimising everyday problems as much as issues from mathematics (e.g. the body of minimal resistance and the Lagrange multiplier) and physics (e.g., the principle of least action by Maupertuis).

In the 19th century, the first optimization algorithms were presented. We recall the least square method by Legendre, which was also claimed by Gauss; Cauchy suggested the gradient method for finding a local minimum of a differentiable function and Fourier formulated Linear Programming for solving problems arising in mechanics and probability theory. Optimisation became a fundamental component of economic theory too.

In the early 20th century, mathematician and economist Kantorovich developed pioneering linear programming techniques, for which he also

won the Nobel Memorial Prize in Economics together with Koopmans. During World War II, military and industrial applications led to improvements in the optimization of logistics, scheduling, and resource allocation (e.g., the application of dynamic programming for reservoir management by Massé).

The milestone of modern optimisation was set by Dantzig[2], an American mathematical scientist who presented in 1947 the simplex algorithm. The simplex is a foundational computational tool in linear programming that, for the first time, was used for practical planning and decision-making in economic applications. Dantzig remarkably foresight envisioned the potential of linear programming implemented in real-world planning challenges. In the 1950s the, non-linear programming was firstly advanced e.g., quadratic optimization by Markowitz, and later the adaptation of the simplex to non-linear programming by Zoutendijk. In the 1990s, with the advent of powerful computing technology, the field of mathematical optimisation expanded with the addition of metaheuristic algorithms (such as genetic algorithms and simulated annealing) and convex optimization techniques. Optimisation also plays an important role in machine learning algorithms, refines and trains the model, improving the accuracy and efficiently navigating the complex parameter space.

2: For reference, Gurobi Optimisation company [143].

5.1.2 Optimisation strategies

Optimization methods can be neatly classified based on two main aspects: objectives and nature. This classification system simplifies the understanding of optimization methods; it is based on the ESTECO documentation compared with the more recent reference by Mashwani et al. [114].

The first aspect, objectives, distinguishes between methods that target a single goal and those that tackle multiple conflicting goals simultaneously. For instance, traditional linear programming falls into the single-objective category, while evolutionary algorithms and genetic algorithms are examples of multi-objective approaches. The second aspect, nature, categorizes optimization methods into deterministic and stochastic types. Deterministic methods operate under the assumption that all input parameters are precisely known, offering exact, deterministic paths to optimal solutions; linear programming and gradient-based methods are examples. On the other hand, stochastic methods, like simulated annealing or genetic algorithms, embrace uncertainty and randomness, aiming for robust solutions that perform well on average.

The interplay between accuracy and robustness is found in the strategies of global and local refinement. By exploring many scenarios in the entire solution space, global refinement sets the foundation for finding solutions that stay strong in various situations. Hence, it may compromise some level of precision in individual regions in favour of a more robust overall

solution. Conversely, local refinement directs its focus towards accuracy within specific regions, but it may not guarantee robustness across the entire solution space. There are four macro categories in which the optimisation strategies can be distinguished depending on the specific problem (see Silveira et al. [172]); they are listed in the following.

1. The primary aim of linear programming (LP) is to find the best outcome in a mathematical model maximising or minimising a linear objective function subject to linear equality and inequality constraints[3]. This technique can be expressed by the following equation
$$\max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in \mathbb{R} \wedge \mathbf{A}\mathbf{x} \leq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}\}$$
where $\mathbf{c}^T\mathbf{x}$ is the objective function that has to be optimised, $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ are constraints and $\mathbf{x}$ represents the variables that have to be determined. The LP is exemplified by the simplex method, designed by Dantzig and named Simplex (see the paper by Nelder and Mead [135]), an algorithm that propels from one feasible solution to another, gradually approaching the optimal one through iterations (see the book by Bloom and Sekhon [14]). The Simplex is efficient for problems with a relatively small number of variables and constraints, its effectiveness is limited for large-scale and more complex problems.
2. The gradient-based approaches are deterministic methods, such as Gradient Descent. They navigate the solution space by following the slope (gradient) of a function, ultimately converging towards the minimum for optimal results. Gradient-based methods rely on the derivative of the objective function (in Arora [6]); rather than scanning the entire solution space, these methods perform local exploration around the current solution. This makes them particularly suitable for problems with smooth and continuous objective functions.
3. The meta-heuristic algorithms, like simulated annealing and ant colony optimization, use higher-level strategies to explore and find optimal solutions. The evolutionary algorithms are a sub-set of meta-heuristic; they include genetic algorithms and particle swarm optimization that draw inspiration from the natural world. They simulate the evolution of a population of solutions, progressively refining and adapting to identify the most optimal solution for a given problem.
4. Dynamic programming takes a unique approach by breaking down complex problems into simpler, overlapping sub-problems.

3: See the presentation by Professor Parrilo P. [146].

5.1.3 Multi-objective optimisation procedure

As mentioned in the book by Miettinen (1998) [118], multi-objective optimisation problems are very common real world challenges, their peculiarity is to entail multiple objectives that need to be optimised

simultaneously. As these objectives may conflict, finding a single "best" solution that optimally satisfies all objectives is not feasible. However, there are trade-off solutions for which no other solution is better in all objectives, these are called non-dominated results and they constitute the Pareto[4] front. Resolving issues with conflicting objectives typically involves the engagement of a human decision-maker, who must express the preference among alternative methods and later on solutions. None of the existing methods for multi-objective optimization can be universally classified as the best for every situation [118], given the multitude of aspects to consider, and many comparison criteria have a somewhat ambiguous nature.

A common approach in multi-objective optimization involves following the steps illustrated in the Fig. 5.2, inspired by the corresponding diagram in Raza et al. (2018) [159].

The first step involves thoroughly framing the problem:

- to identify the objectives for the optimization process, determining whether the goal is to maximize a certain outcome, such as profit or efficiency, or to minimize something, like cost or error;
- to establish constraints, any limitations or restrictions needed to validate the problem;
- to determine input and output variables along with their ranges.

The resulting dataset is referred to as design space; it defines the boundaries within which the exploration of inputs is allowed. When this exploration is done systematically, an appropriate technique is implemented, which is called Design of Experiments (DOE). In the third step, the objective values within the design space are utilized to construct a surrogate model, providing an approximation of the objective function. The objective functions are mathematical formulations that measure the performance or desirability of a specific outcome, considering the values of input parameters. The next important step involves selecting a suitable optimization method, a choice driven by the nature of the current problem. Following this, the goal is to locate the Pareto optimal front. It is crucial to evaluate the validity and effectiveness of the optimization results. Does the chosen optimal solution exist in the design space? Does it meet the constraints? If the solution proves unsatisfactory or there is room for improvement, the optimization process becomes iterative: the model is refined by adjusting the parameters, and then the process is repeated until an acceptable solution is reached. When this happens, the multi-objective optimization process concludes.

4: For reference, Britannica Encyclopaedia [21]. Between the nineteenth and twentieth centuries lived Vilfredo Pare, an Italian economist and sociologist. His intellectual pursuits led him to contribute significantly to diverse fields, bridging the realms of philosophy, politics, and mathematics in his quest for understanding and analysing economic phenomena. In studying economic systems, Pareto demonstrated that the distribution of incomes and wealth in society was not equally allocated. He introduced the idea of the Pareto Optimum, arguing that an allocation is Pareto optimal when you can not make one person better off without making another worse off. The evolution of Pareto's ideas continued beyond his time; in the context of multi-objective optimization established the concept of the "Pareto front."

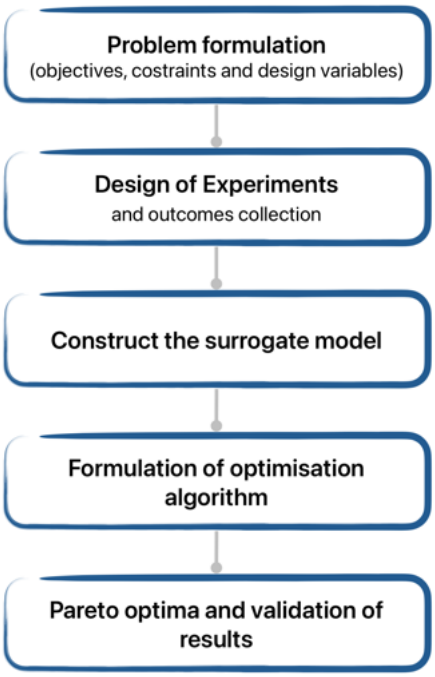


Figure 5.2: Multi-objective optimization procedure (based on [118]).

5.2 Design of Experiments

The Design of Experiments, abbreviated by DOE, is a systematic and structured approach used in scientific research and industrial settings to plan, conduct, and analyse experiments. The primary goal of DOE is to gather relevant data efficiently and extract meaningful insights about the factors that influence a particular process or system. As extensively explained in the book by Montgomery (2012) [130], by strategically designing experiments, researchers can identify critical variables, optimize conditions, and improve the efficiency and quality of processes. DOE is widely applied in fields such as engineering, manufacturing, product development, and various scientific disciplines. The DOE is required to initialize the optimization algorithm; it is important in delineating the key design variables, determining methods for their reduction, understanding their interrelationships, and establishing a reasonable number of objectives and constraints for the optimization project.

Experimental design is founded on three key principles that are randomization, replication, and blocking. By introducing randomness into the assignment, researchers aim to minimize the influence of potential extraneous factors. The purpose of replication is to estimate the variability within each treatment group; indeed, having multiple instances of each treatment enhances the precision of the experiment's results. The principle of blocking involves grouping experimental units that share similar characteristics. By doing so, researchers can control for variability within these characteristics; this ensures that any observed differences among treatments are not mistaken by variations in the characteristics used for blocking.

In DOE, the factor is the independent variable subjected to variation, while the level represents the extent to which a factor has been modified; it can be expressed in discrete or numeric terms. Before conducting an experiment, it is important to know exactly what is being studied, how data will be collected, and to have at least a basic understanding of how the data will be analysed. This could be referred as pre-experimental planning. When considering factors that might affect a process or system's performance, the experimenter usually categorizes these factors as either potential design factors or nuisance factors. This categorization helps guide the selection of factors, levels, and the range to be explored in the experiment.

Based on [130], the modern evolution of statistical experimental design can be categorized into four eras, which are summarized in the list below.

1. From the paper by Dwornicka and Pietraszek (2018) [55], we know that the first work on DOE dates back to 1921 by Fisher [66], who is also regarded as the father of modern statistics and experimental design, and the word DOE was first used as the title to the book by Fisher in 1935 [67]. In his studies, Fisher introduces the analysis of variance (ANOVA), which later on effectively developed in the Latin Square (LS) scheme. The LS was extended to form the Full Factorial (FF) experimental design.
2. The initial applications of DOE were in the field of agriculture, however, its significance quickly expanded to industry, including manufacturing. The industrial era was accelerated with the introduction of Response Surface Methodology (RSM) by Box and Wilson in 1951 [18]. They noticed that industrial experiments differ from agricultural ones in two main ways: the production is faster, and information related to the products can quickly be gathered from a small set of runs.
3. The third era is characterized by the contribution of Genichi Taguchi in the late 1970s (see the book by Phadke (1989) [148]). Referring to his proposal as robust parameter design, with the aim of minimizing variability around the target value to prevent quality loss in industrial production and making processes less sensitive to environmental factors and products variation (e.g., material ageing), Taguchi's concept of utilizing DOE gained popularity in the industry, such as automotive and aerospace manufacturing, electronics, semiconductors, and various other sectors [130]. The Taguchi method was originally introduced as a practical implementation of factorial designs.
4. Nowadays, the advancements in computer software have greatly simplified the implementation of DOE techniques. The utilization of designed experiments has expanded well beyond its agricultural roots; every field of science and engineering has effectively incorporated statistically designed experiments.

5.2.1 DOE techniques

Some of the most common DOE methods are included in this section.

In the book by Lawson (2015) [102], the purpose of experimentation is to examine the sources of variability in the response, based on the variability of the factors. Studying variability sources has several reasons: to gain more insight into variance sources and propose new ideas for future experiments, to find ways to reduce variations in the response. When the analysis involves a single factor and its levels, it is referred to as ANOVA. In ANOVA, sums of squares, mean squares, and degrees of freedom of responses are considered. Additionally, it is emphasized that the experiment should be conducted in a random order to ensure uniformity in the environment where the treatments are applied [130]. The Latin square method was developed to enhance the efficiency of ANOVA design. Since It features two separate blocking factors the possibility to analyse the variance of two factors simultaneously enhances the precision in detecting differences among responses. However, it comes with limitations: only three factors can be analysed (in this case, the sample is named Latin cube), the number of levels must be consistent across all factors, and interactions between factors are not permitted [55].

When the experiments are more complex and involve two or more influential factors, one approach is to explore every possible combination of the factors and their respective levels. This approach is known as factorial design and offers two primary advantages: it allows for a better understanding of interactions or joint effects among the factors; it improves experiment efficiency, implying that the same results can be obtained with fewer experiments compared to applying ANOVA to each individual factor [102]. However, in numerous scenarios, it is impossible to perform all of the runs included in a factorial DOE under homogeneous conditions. Hence a more viable approach is to execute only a fraction of these runs. In fractional factorial design, only a subset of experimental runs from the full factorial design are systematically selected, not all the combinations are accounted for. As the response is typically influenced by few main effects and lower-order interactions, with higher-order interactions often proving insignificant, fractional factorial designs can spare time and resources (from the paper by Durakovic [54]). Factorial methods are particularly advantageous in the early stages of experimental research when many factors are still partially unknown and need investigation.

As confirmed by Lin and Tang (2014) [105], as computer power increased and experimenting with it became easier, the number of potential runs significantly grew, and space-filling methods became widespread. Experiments determined by space-filling methods are recognized for exploring a bounded design region as uniformly as possible, effectively capturing the system's behaviour across diverse conditions.

Based on the Latin square concept, McKay et al. in 1979 [116] proposed a space-filling sample design named Latin hypercube sampling (LHS), a multidimensional distribution for analysis of three or more factors (in Fig. 5.3 a comparison between factorial and LHS from the 2021 paper by Choi et al. [29]).

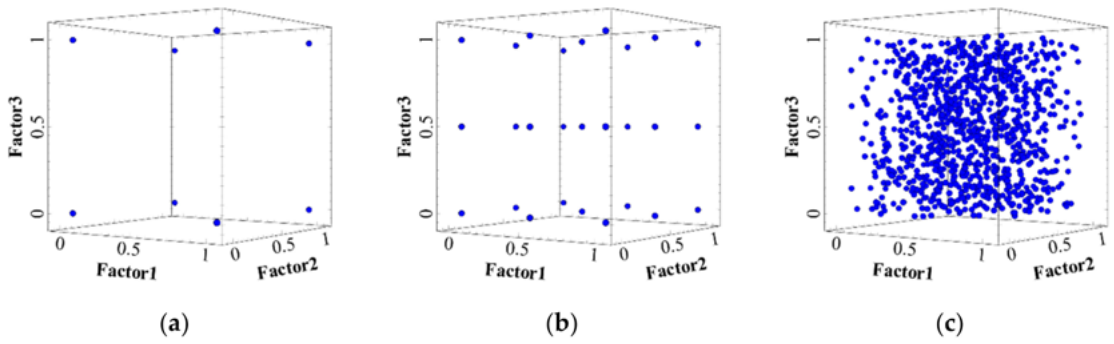


Figure 5.3: Comparison of data sampling methods in [29]. (a) Two level full/fractional factorial design; (b) three level full factorial design; and (c) Latin hypercube sampling design.

The LHS is an advanced form of Monte Carlo Sampling, the independent pseudo-random generation of numbers according to a PDF. More precisely, in LHS, the design space is partitioned into a finite number of equi-probable intervals, and a single random value is chosen from each interval, creating a well-distributed set of points (from the paper by Blatman et al. 2007 [13]). The primary aim of this method is to reduce correlations among input variables and maximize the separation between the generated designs. Another space-filling technique is the Sobol sequence (from Sobol 1967 [173]), which is a low-discrepancy (or quasi-random) sequence of samples. The experiments in Sobol consist of deterministic samples characterized by low discrepancy, indicating that the deviation of the set from a uniform distribution is minimal and can quickly converge (further details in the 2021 work by Renardy et al. [161]). In Fig. 5.4, we see a comparison between a random distribution, an LHS, and a Sobol sequence; these methods are especially well-suited for initiating an optimization procedure.

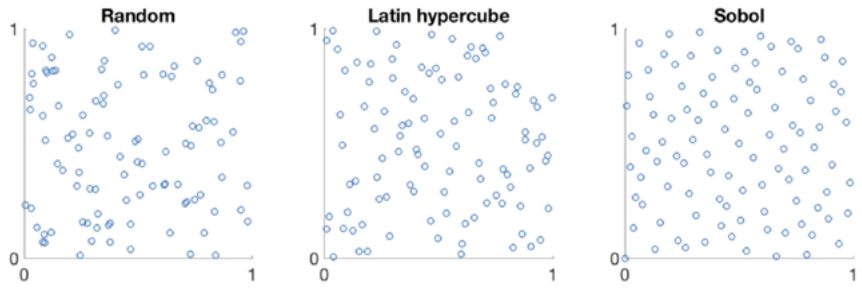


Figure 5.4: Comparison between random sampling, LHS and Sobol sequence by [161].

5.2.2 Response Surface Methodology

Within the realm of DOE methods, one can find the Response Surface Methodology (RSM), a comprehensive collection of statistical techniques and analysis tools employed for multi-factor experimental design and multi-objective optimization [130, 102, 55].

From the modeFRONTIER guidelines we know that, while high-performance computing enables accurate experiments, replicating real-world phenomena often requires time-consuming simulations. Engineers employ DOE to reduce calculations and use RSM tools to analyse and interpolate outcomes, creating a meta-model for predicting responses. Indeed, the main goal of RSM is to model and optimise the connection between one or more independent factors and a desired response. In RSM, both the factors and the response typically are continuous numerical data. The interconnection between factors and response is approximated using regression models, generally polynomials (e.g., using least square method), obtaining response surfaces. The RSM predictive model is employed to estimate (many) fresh unknown outputs without the need for additional computationally expensive simulations. These new outputs are commonly referred as virtual because they are not computed by a real solver.

RSM experiments are usually utilized in the final stages of experimentation when key factors have already been identified and investigated. RSM is also used for optimization, where by adjusting the input factors, one can identify the maximum, minimum, or target values among the virtual responses. The accuracy of the RSM model is measured based on the difference between the predicted and the true responses. This difference is called prediction error or residual. If the residual is higher than an arbitrary threshold value, the RSM must be trained. The training consists of running additional designs and iteratively comparing until convergence is achieved. In multi-objective optimisation, the trade-offs between objectives are non-dominated solutions in the Pareto front.

5.3 Optimization of the moon pool with modeFRONTIER

The capabilities of mF enable mathematical optimisation of parametric processes, as it is the moon pool case. Examples of theoretical and practical works can be found respectively in Clarich et al. [30] and Russo et al. [166]. In the upcoming sections, we will outline not only the procedures we followed to reach this goal, and present the fundamental principles of the software. Our sources include optimization literature and the software library, which are accessible to any user upon license application.

Recently, attention has been drawn to the moon pool recess, emphasizing its role as a damping device, particularly in open-sea activities. It was observed even in Chapter 4 that the recess consistently influences the flow dynamics within the pool. Grounding our work on the benchmark by Guo et al. [79], we aim to minimise the wavy behaviour of the flow inside the moon pool optimising the shape of the recess. At the end of the analysis, we expect to be able to define a trend for the recess shape on the basis of the flow conditions (i.e., in our study, the frequency of the sloshing).

The constitutive environments in mF are four: the *Workflow* for planning the optimization process, the *Planner* in which the algorithm(s) for the Design of Experiments (DOE) method and the optimisation techniques are selected, the *Run Analysis* to run the simulations and check real-time outputs, the *Design Space* where data from simulations are reprocessed.

5.3.1 First step: the Workflow

The first step to initialize an optimisation in mF consists of the Workflow environment, which is a canvas where the user creates the working process. The building blocks of the Workflow are called nodes; they are allocated with variables (scalar variables or vectors), actions or functions; nodes are connected through links that define the direction followed by the information. Basically, the Workflow comprises two fundamental parts: data flow and process flow. This is why, mF is also equipped with an embedded Python library for programming and a terminal environment for commands. In addition, probably the most significant aspect is the option to integrate third-party software assigned to compute data, like run simulations. Indeed, there are nodes that integrate third-party software for CAD and CAE applications.

In general, the user sets up the nodes for the input parameters, which are the independent variables of the problem, specifying the range of variation and the unit measurement per each. Inputs are linked to the process-part of the Workflow, such as the node for simulations. The simulation node is then followed by the nodes destined for the output. The process is bounded by *Start* and *Exit* nodes, in between the user can add as many functional nodes as needed to finalise the process.

Fig.5.5 represents the workflow that has been devised for optimizing the moon pool. The first input is the selected set of frequencies $f = [0.47; 0.71; 0.79; 0.87; 0.93; 1.01; 1.18; 1.75]$ Hz, that will be implemented to generate the sloshing. These frequencies have been used in [79] to validate the model to calm water. The size of the two-dimensional recess is the second input, the length R_L and height R_H of the recess are encapsulated in two different nodes; since the mean value encompassed in a reference range is requested for any variable, to the recess has been given the equivalent size of the benchmark [79]. The simulations are

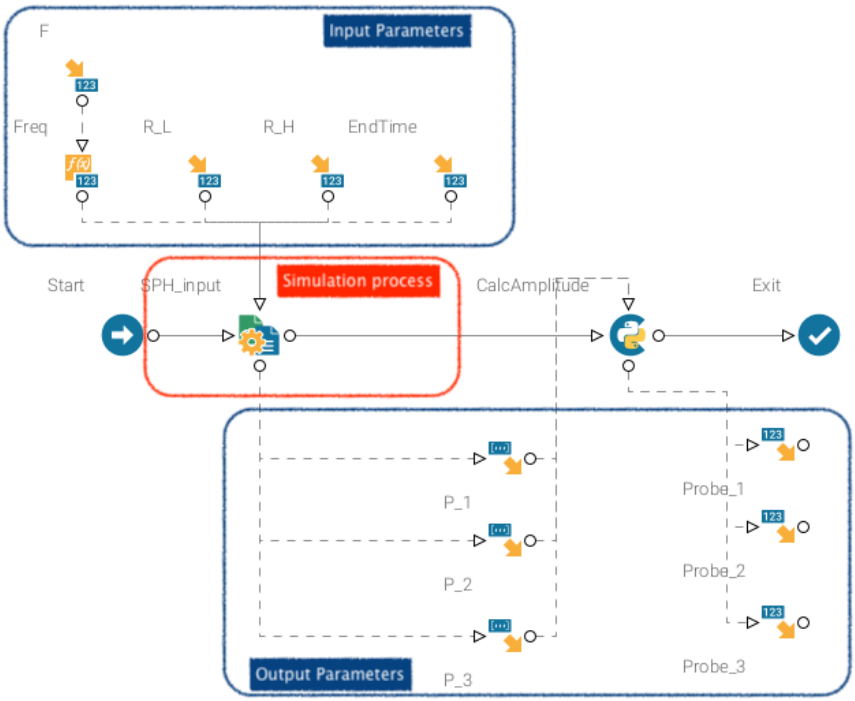


Figure 5.5: Simplified workflow designed for the moon case, the fundamental parts are: input parameters, simulation process and output parameters.

called by mF process node, computed by our in-house code on the remote machine powered by the Nvidia GPU (representation in Fig. 5.7). The output data are sent back and stored in tables in mF, where they are ready for post-processing (that is a job for the Design Space environment).

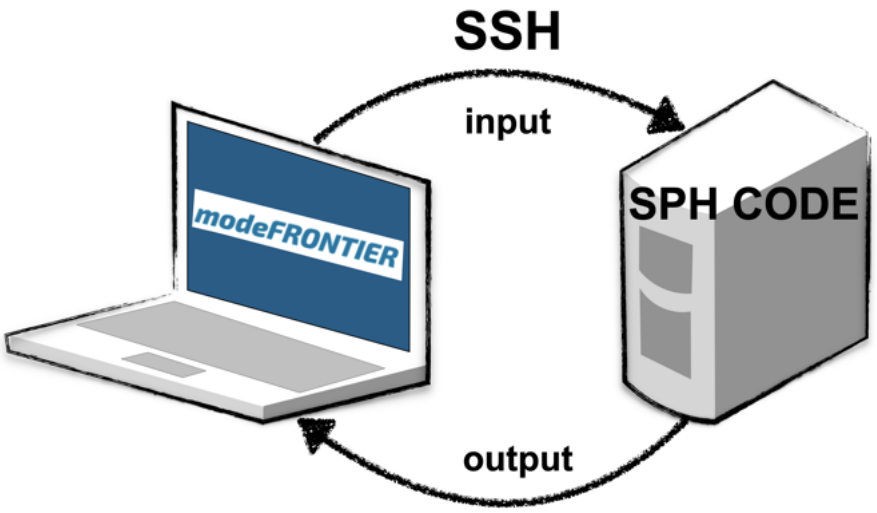


Figure 5.6: The process in mF: input data are sent to the computing machine, at the end of the running results are collected and stored back in mF tables.

5.3.2 Second and third steps: the Planner and the Run Analysis environments

When the workflow is ready, the user opens the Planner, the environment assigned to initiate the DOE strategy and the optimisation.

The fundamental aspect is to have a clear understanding of the objective that needs to be achieved; thus, pre-experimental planning plays a vital

role in establishing the entire procedure. Determining the optimization method is important, and it depends on the number of objectives to be optimized and the available number of runs. In this investigation, our goal is to minimize the wave height within the moon pool, which is determined by the water level detected by three probes on the side walls and in the center of the pool. As illustrated in Fig. 4.8b, we will look for the highest and the lowest level of water detected at the probe P_i position in a defined time range. The objective is to minimise the difference $\Delta H_{wave}^{P_i}$ between these two values. We have three objectives, meaning the three probes, and it is uncertain whether they conflict. Additionally, due to computational expenses, we cannot afford to run numerous simulations. As suggested in mF guidelines, a practical choice is to opt for optimization involving a DOE explorations of the parameters and the creation of a meta-model using the RSM: with RSM, we can effectively work with a limited number of real simulations and numerous virtual ones.

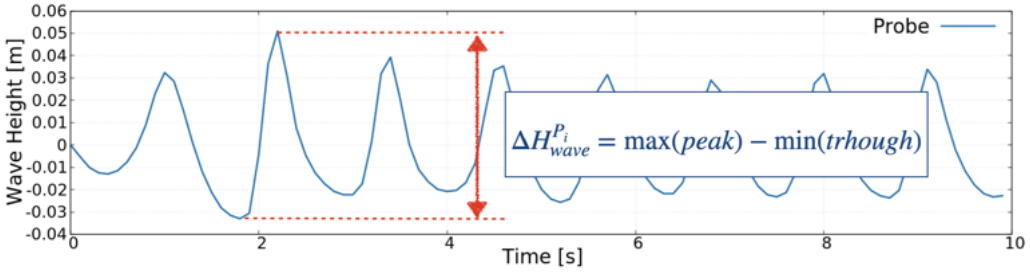


Figure 5.7: Representation of the objective of this investigation: minimise the value $\Delta H_{wave}^{P_i}$ which is the difference between the highest max($peak$) and the lowest min($through$) wave level in a given time range, at any probe position. The time range is a trade-off value that assures the reach of the oscillatory behaviour of the flow requestion a feasible computational effort.

The initial phase involves designing the experiments, which includes defining input and output variables and specifying their respective ranges. The input parameters for this study consist of the frequency of the external force, the recess length and the recess height. The frequency values are taken from the set mentioned in section 5.3.1, while the recess dimensions are considered within a specified range. The choice of these ranges is influenced by the research conducted by Hammargren and Törnblom [81] on drag reduction based on the moon pool height. Given that the vessel type studied in [81] closely resembles the one proposed in Guo et al. [79], we have adopted the same upper limit as mentioned in [81]. Moreover, in [81], they explored a range of R_H and found that the mid-value was more conservative in terms of drag reduction compared to the boundary values. The authors demonstrated that an optimal value exists for the R_H. The ranges of the recess measures are:

- the recess length R_L varies between 11% and 45% of the upper opening of the moon pool, which in the scaled model corresponds to $0.1 - 0.41$ m;
- the recess height R_H varies between 15% and 48% of the hull depth,

which in the scaled model corresponds to $0.05 - 0.163\,\mathrm{m}$;

assuming that wave response to R_L variation is comparable to the one by R_H. We also question whether the noteworthy observation by [81] holds true under different sea conditions. Thus these ranges will remain unchanged and will be tested per each frequency. To sum up: because of the assumed variability range of geometric parameters R_L and R_H that are sought to be further optimised, the present task represents, in fact, a constrained optimisation problem.

The next step is deciding which DOE is the most appropriate technique. Given that we have three factors, the three input parameters, each with multiple levels, the Latin Hypercube seems like a suitable choice to uniformly explore the design space. We begin with a relatively small number of 15 runs. During the DOE process, mF manages the entire procedure: mF initiates the first simulation remotely on the running machine, and once completed, the results are stored in a table in mF. This process is then automatically repeated. As SPH simulations are well-known for being time-consuming, increasing the time step can accelerate the process, at the cost of modifying the physical model. As the statistical (RANS-type) turbulence modelling has not been included in our in-house SPH solver, and the LES model would be prohibitively time consuming for 3D simulations, for the purpose of the present optimisation task we assumed higher and fictitious values for the viscosity of the phases. Respectively, we have $\mu_0^I = 10.0\,\mathrm{Pa/s}$ for the heavier phase and $\mu_0^{II} = 0.1\,\mathrm{Pa/s}$ for the lighter phase. The reference density of the heavier phase is $\rho_0^I = 1000.0\,\mathrm{kg/m^3}$, while for the lighter phase is $\rho_0^{II} = 1.0\,\mathrm{kg/m^3}$. The trade-off resolution is characterised by a smoothing length of $h = 0.015625\,\mathrm{m}$ (further details about are in section 4.2.5). To reach the endtime, which for convenience is taken equal 10 s in simulation time, the machine takes from three to five hours of real time.

In the Run Analysis environment, it is possible to monitor the real-time progress of simulations and other calculations developments, including identifying issues promptly.

Following the finishing of the DOE, the RSM is trained to check whether the residuals exhibit excessively high values, if so a new set of simulations are run to get additional outcomes. We used the Incremental space-filling method, whose algorithm adds new points sequentially by maximizing the minimum distance from the existing points (from mF literature).

Once the RSM is validated, the Planner environment allows for configuring the conditions for optimization, including objectives and the optimization algorithm.

5.3.3 Fourth step: data analysis within the Design Space environment

The outcomes are stored in tables within the Design Environment environment, where they can be organized for statistical analysis and presented through charts and graphs. In this environment, users have the capability to employ Pareto to search for non-dominated results, compare solutions, establish factor correlations, train the RSM, and perform various other actions.

The iterative training of the RSM led us to have 60 adequate runs. Before proceeding with the RSM training, it is a good practice to conduct a sensitivity analysis to identify and exclude the most irrelevant results (outliers): they can lead to the distortion of the response surface function.

The software mF offers a variety of algorithms or tools for the creation of the RSM; they can be usefully compared during the training. These algorithms can be distinguished in interpolating or approximating RSM. Below there is an overview of the algorithms that we have utilized and their key characteristics. The interpolating ones are:

1. Kriging (KR) is a Bayesian method named after Professor Daniel Krige, who developed it for making predictions in geostatistics. This approach is commonly used for predicting values such as soil permeability, extraction of oil, mineral deposits, and more. The KR algorithm functions as an interpolating method, but in the presence of noisy problems, it transitions into an approximating approach. As in the 2022 paper by Kumar et al. [98], it is a statistical RSM that treats the deterministic response as a stochastic process: KR is governed by the covariance function (called variogram) computed using the spatial correlation among the sample points, the approach is to estimate the response at a given point by assigning greater importance to nearby sampling points and then minimise the variance of the residual errors. This procedure is also represented in Fig. 5.8d by [98]. This method's results are rather slow. Thus, it is not suitable in case of a very large datasets.
2. Anisotropic Kriging (AKR), derived from standard Kriging, acknowledges the mutual effect that input variables have between each other. The variogram considers both the distance between training points, usually called lag, and their positions relative to each other in the design domain, referred as vector direction (from mF guidelines).
3. DACE Kriging (DACE), or Design and Analysis of Computer Experiments, was first presented in 1989 by Sacks et al. [169] as a modified Kriging method to create meta-model given multidimensional design space, which was only possible with the advent of computer experiments. As explained in the recent review by Balaban [7], this hybrid method combines regression analysis and Kriging. More

precisely, a polynomial regression model is initially used for the outputs, followed by applying basic Kriging to the residuals.

4. Shepard-K-Nearest (KN) is a statistical interpolation technique that estimates the value of unknown samples based on the weighted average values at known locations; precisely, values closer to the target location have more influence. The weights are determined by the inverse of the distances between the target location and the known points (from the article by Poles in [150]). The training done by KN is not computationally intensive, and, therefore, suitable for large datasets (from mF guidelines).
5. Radial Basis Functions (RBF) can generate interpolating response surface passing exactly through the training points, but only in case of noise-free samples. hey are enabled to approximate general non-ordered datasets by implementing regularization techniques. RBF were firstly presented in 1971 by Hardy [83], and are, by definition, mathematical functions that depend on the radial distance from a reference point. As illustrated in Fig. 5.8b by [98], these methods compute the product of a set of weighted coefficients with the basis functions centred in points that are conveniently selected in the sample space to get the response function (more details in the 2016 paper by Majdisova and Skala [111] and in the tenth chapter by Colaço and Dulikravich in *Thermal Measurements and Inverse Techniques* [33]). RBF functions can be classified in two categories, where the first category includes globally supported while the second locally supported functions, based on the 2021 review by Shah at al. [171]. An example of creating an RSM with RBF is also included in the 2007 paper by McDonald [115]. In mF, the surface fitness is tested by implementing the leave-one-out methodology; one training point is excluded, and the remaining are used to generate the RSM: the quality of the RSM improves as the difference between the two values decreases (from mF guidelines).

In the approximating models, the RSM function coefficients are computed by minimising the error estimated at the data training; the algorithms are:

1. Smoothing Spline ANOVA (SS-ANOVA) is a statistical modelling technique based on function decomposition and it is derived from classical ANOVA and the associated notions of main and interaction effects (additional details in mF guidelines). An example of SS-ANOVA is in Fig. 5.8c by [98] where there are two separate sets of correlated samples, each of them is attributed with an observation point that can be obtained by the least square interpolation with weights according to sample distance. SS-ANOVA is as an effective screening method for identifying crucial variables within a given dataset.
2. Gaussian Processes (GP) are regression techniques that use the statistical inference method, called Bayesian inference, to update

the dataset trend estimation (more details in the 2005 book by Rasmussen and Williams [158]). The GP generates a model that uses means and variances to predict a Gaussian probability distribution rather than a single scalar value as in usual regression models. The GP are considered approximating methods, except when the randomness of the dataset is null, and therefore, they perform as interpolating RSM. By the mF guidelines, these algorithms are best suited for non-polynomial responses.

3. Neural Networks (NN) is a set of statistical and learning models inspired by the brain structure: the computing nodes represent the real neurons, and they are interconnected to mimic biological neural networks (from mF guidelines, but for more exhaustive study, refer to the book by Haykin [85], firstly published in 1999). In the network, the information is stored in synaptic weights, which define the connection strength between neurons activated during training and prediction. Generally, the NN process comprises two phases: the feedforward NN, in which the information travels from input layer, hidden layer and output layer; the back-propagation phase, in which the evaluation of the residuals determines the accuracy of the RSM. The network weights are adjusted appropriately to minimize the mean square error revealed by the residuals.
4. Stepwise Regression (STEP) is a relatively simple approach; large data are split into smaller datasets and then approximated.
5. Polynomial Singular Value Decomposition (SVD): based on the definition in the 2020 book by Blum et al. [15], the SVD of a matrix is the factorisation of this matrix into the product of three other matrices: a complex unitary matrix, a rectangular diagonal matrix with non-negative real numbers on the diagonal and another complex unitary matrix, that reveals the underlying structure of the original matrix. This mathematical tool can be applied to regression analysis (see the paper of 1982 by Mandel [112]), and based on the notes by Fernandez-Granda C.[5], the SVD of the covariance of a dataset shows how the data direction vary in the design space. Polynomial SVD produces robust polynomials of the main dataset trends by minimizing the squared error predictions of the dataset.

5: For reference, SVD [63].

The RSM training is carried out individually for each objective in mF as follows. The final dataset of run is randomly divided into two subsets, with the larger portion (approximately 80%) used for creating the meta-model and the remaining part allocated for validation. The validation set values are then estimated to compare virtual and real outcomes: the resulting residuals are analysed and presented in a table as the one in Fig. 5.9a, featuring a comparison of various error types. The optimal algorithm, the one that has the lowest residual value, is automatically indicated in mF. The list of errors are generated by the algorithm during the training phase; they are used to evaluate the accuracy of the RSM. The main error values to check with are:

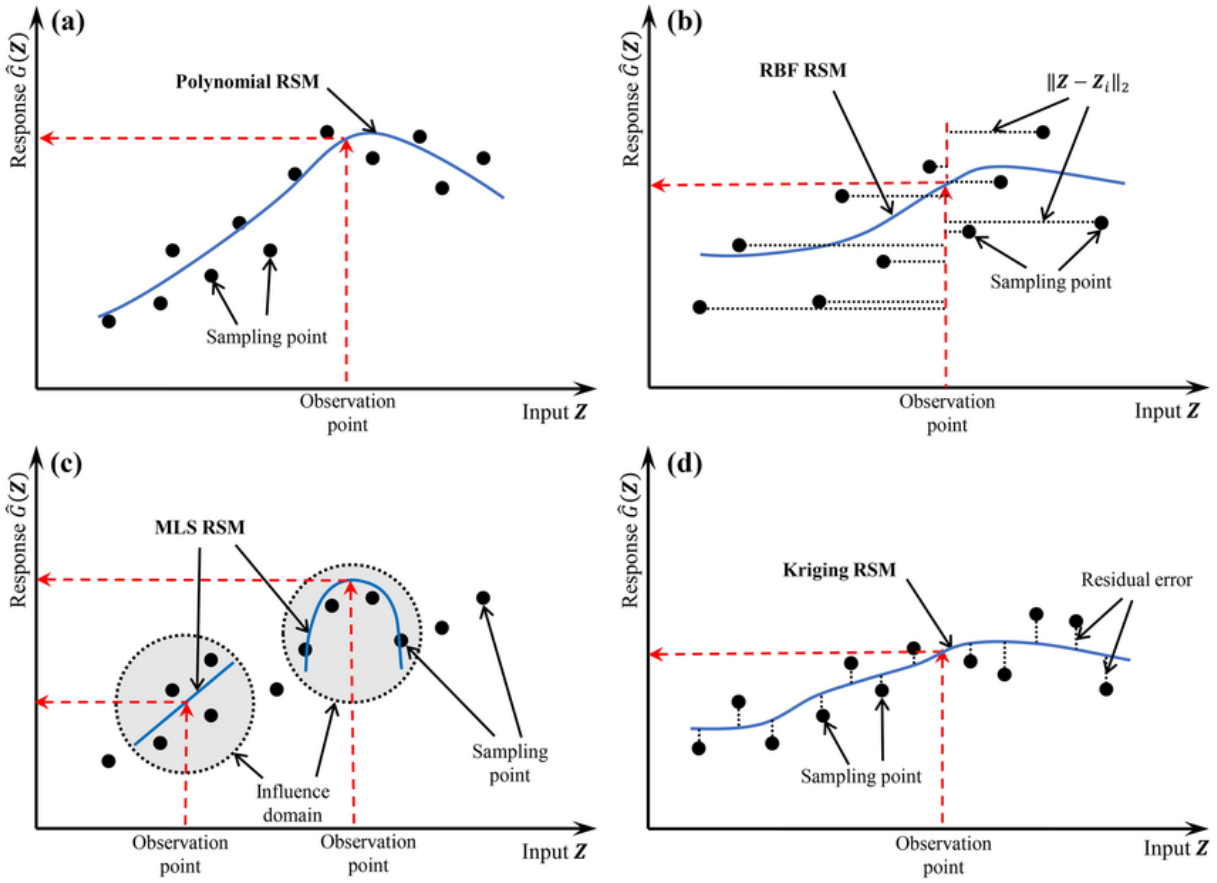


Figure 5.8: Partial illustration taken from [98]; function approximation to generate RSM: the response corresponding to a specific input, called observation point, is obtained by manipulating the experimental sampling points. Input variables in abscissa, and the corresponding response value is in ordinate. (a) Polynomial RSM, e.g. SVD: simple and robust method, the observation point response is approximated with polynomial using sample values. (b) RBF RSM: the norm $||Z - Z_i||$ between the observation point and the sample values is multiplied by a weight and the radial basis function centred within the observation point. (c) Moving least square RSM, e.g. SS-ANOVA: are defined two sets of samples, each of them is attributed with an observation point that the least square interpolation with weights can obtain according to sample distance. (d) Kriging RSM: estimating of the response at the observation point by assigning greater importance to nearby sampling points and then minimise the residual error variance.

- the mean absolute error, which is the average value of the difference between real and virtual designs;
- the mean relative error is the percentage average obtained by the ratio between the mean absolute error and the real design values;
- the mean normalized error which is the normalisation of the mean absolute error based on the percentage absolute value of the output variability in its given range;
- the R-squared coefficient of the RSM regression;
- when the meta-model is created using data, it never exactly aligns with them, leading to loose information when employing the model for subsequent tasks. The Akaike Information Criterion, or AIC, estimates the relative quality of the model by measuring the relative gauge of the information lost during the model creation. When presented with a set of models for the data, the quality of each model is evaluates using the AIC. In comparison to the others, serving as a tool for model selection, the lowest AIC correspond to

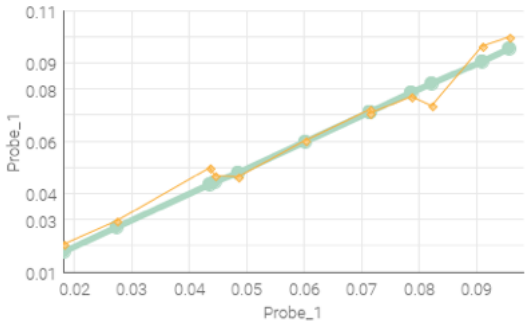
the optimal model.

In addition, the validation is represented through a plot similar to the one shown in Fig. 5.9b; each algorithm has its specific plot. In this plot, both axes pertain to the wave response at the chosen probe position: the green line represents the validation results obtained from real simulations, while the yellow scatter line links their corresponding RSM estimations. The accuracy of the RSM can be visually assessed with smaller residuals resulting in a yellow line that aligns more closely with the green line.

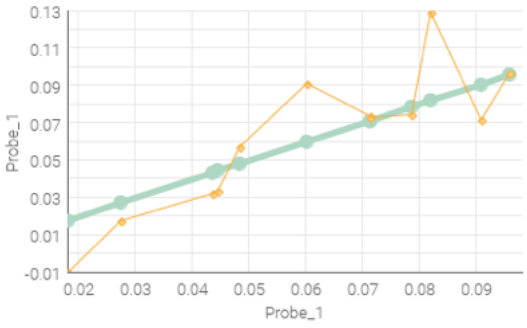
The three sets, Fig. 5.9, Fig. 5.10 and Fig. 5.11, show the errors table, the validating plot of the best fitting algorithm (on the bottom left corner) and the plot of a worse performing algorithm (on the bottom right corner), respectively for probe P1, P2 and P3.

Output	Name	Mean absolute error	Mean relative error	Mean normalized error	R-squared	AIC
Probe_1	Probe_1_AKR_0	5.89E-3	1.32E-1	7.58E-2	8.97E-1	-1.11E1
Probe_1	Probe_1_DACE_0	5.29E-3	1.17E-1	6.81E-2	9.13E-1	-1.71E1
Probe_1	Probe_1_GP_0	5.73E-3	1.45E-1	7.38E-2	8.92E-1	-1.25E1
Probe_1	Probe_1_KN_0	8.83E-3	2.32E-1	1.14E-1	7.62E-1	-1.03E2
Probe_1	Probe_1_KR_0	1.09E-2	2.16E-1	1.40E-1	6.56E-1	-5.44E-1
Probe_1	Probe_1_NN_0	1.46E-2	3.34E-1	1.88E-1	2.98E-1	-3.20E1
Probe_1	Probe_1_RBF_0	3.11E-3	6.23E-2	4.01E-2	9.72E-1	-3.28E1
Probe_1	Probe_1_SS-ANOVA_0	7.33E-3	1.37E-1	9.44E-2	8.44E-1	-1.40E1
Probe_1	Probe_1_STEP_0	1.52E-2	2.96E-1	1.96E-1	3.34E-1	-8.46E1
Probe_1	Probe_1_SVD_0	1.24E-2	2.40E-1	1.60E-1	5.97E-1	-9.26E1

(a) RSM training, error comparison of algorithms; the smaller residuals are given by the RBF algorithm.



(b) RBF - best fitting



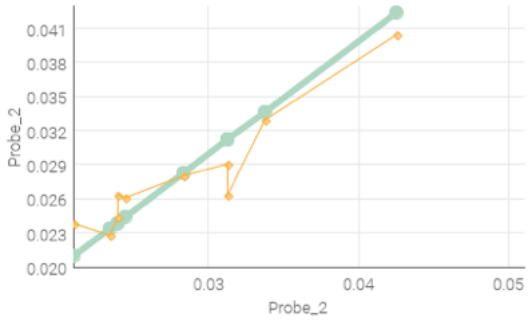
(c) NN - bad fitting

Figure 5.9: RSM training of Probe 1: (a) error comparison of RSM algorithms; (b) the best fitting algorithm; (c) example of bad fitting algorithm.

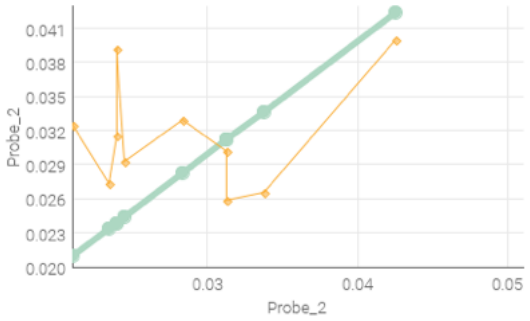
Despite three algorithms DACE, GP and KN are suggested as optimal in Fig. 5.11a, we opt for the DACE RSM that has the lowest mean normalised error. In our study case, we achieved better fitting by constructing the RSM using interpolating methods rather than approximating ones: RBF for both, the first and the second probes, and DACE for the third probes. This could be attributed to the fact that the simulation dataset is not exceedingly large, making interpolation yield more accurate results. These optimally fitting algorithms will be employed in the subsequent stages of the optimization process.

Output	Name	Mean absolute error	Mean relative error	Mean normalized error	R-squared	AIC
Probe_2	Probe_2_AKR_0	3.49E-3	1.37E-1	1.63E-1	4.69E-1	1.84E0
Probe_2	Probe_2_DACE_0	3.05E-3	1.13E-1	1.42E-1	5.83E-1	-4.59E0
Probe_2	Probe_2_GP_0	3.50E-3	1.36E-1	1.63E-1	4.75E-1	-2.80E-1
Probe_2	Probe_2_KN_0	2.84E-3	1.14E-1	1.32E-1	6.25E-1	-1.08E2
Probe_2	Probe_2_KR_0	5.02E-3	1.86E-1	2.34E-1	1.60E-1	2.42E0
Probe_2	Probe_2_NN_0	6.03E-3	2.15E-1	2.80E-1	-5.31E-1	-3.56E1
Probe_2	Probe_2_RBF_0	1.79E-3	6.49E-2	8.32E-2	8.68E-1	-1.81E1
Probe_2	Probe_2_SS-ANOVA_0	2.77E-3	1.04E-1	1.29E-1	7.31E-1	-1.70E1
Probe_2	Probe_2_STEP_0	4.10E-3	1.52E-1	1.91E-1	3.83E-1	-9.47E1
Probe_2	Probe_2_SVD_0	6.39E-3	2.51E-1	2.98E-1	-5.15E-1	-8.97E1

(a) RSM training, error comparison of algorithms; the smaller residuals are given by the RBF algorithm.



(b) RBF - best fitting



(c) SVD - bad fitting

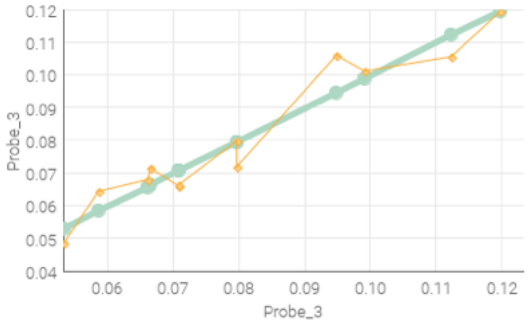
Figure 5.10: RSM training of Probe 2: (a) error comparison of RSM algorithms; (b) the best fitting algorithm; (c) example of bad fitting algorithm.

The multidimensionality of the RSM for the moon pool case prevents from having a single figure in which visualize the effect of all the three input parameters on the response surface simultaneously. An alternative approach is to examine the surface by selecting two inputs at a time per each probe: given a constant value of R_H, display the frequency and the R_L plot; given a constant value of R_L, display the frequency and the R_H plot.

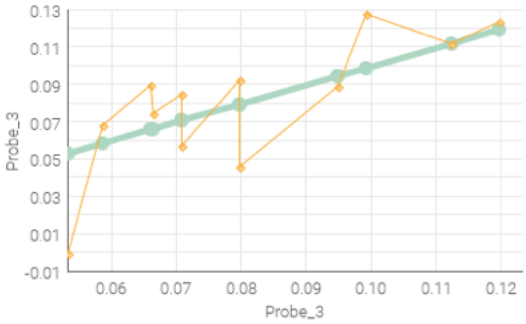
Regarding the first probe, in Fig. 5.12 we see a comprehensive representation of the 3D plot of the response model, Fig. 5.12a at constant $R_H = 0.1068$ m, and Fig. 5.12b at constant $R_L = 0.2559$ m. When changing manually the third constant factor, for example the value of R_H in Fig. 5.12a, we can have an overview of the effect of the three input factors altogether. Each sub-figure consists of four panels with 2D plots of the quantities that are analysed, the wave behaviour at the probe position for variable frequency and the recess measure (R_L and R_H respectively). Regarding the second probe, in Fig. 5.13 we see a comprehensive representation of the 3D plot of the response model, Fig. 5.12a at constant R_H and Fig. 5.12b at constant R_L respectively. Finally, the first probe, in Fig. 5.14 we see the 3D plot of the response model detected from the third probe.

Output	Name	Mean absolute error	Mean relative error	Mean normalized error	R-squared	AIC
Probe_3	Probe_3_AKR_0	4.72E-3	6.16E-2	7.12E-2	9.25E-1	-1.87E1
Probe_3	Probe_3_DACE_0	4.62E-3	6.08E-2	6.96E-2	9.25E-1	-2.27E1
Probe_3	Probe_3_GP_0	4.68E-3	6.44E-2	7.05E-2	9.28E-1	-2.11E1
Probe_3	Probe_3_KN_0	6.63E-3	8.21E-2	9.98E-2	8.41E-1	-1.12E2
Probe_3	Probe_3_KR_0	1.41E-2	1.86E-1	2.13E-1	1.31E-1	6.72E0
Probe_3	Probe_3_NN_0	9.83E-3	1.36E-1	1.48E-1	6.62E-1	-4.46E1
Probe_3	Probe_3_RBF_0	6.94E-3	8.22E-2	1.04E-1	7.50E-1	-1.02E1
Probe_3	Probe_3_SS-ANOVA_0	5.42E-3	6.95E-2	8.16E-2	9.03E-1	-2.35E1
Probe_3	Probe_3_STEP_0	1.74E-2	2.51E-1	2.61E-1	-2.56E-1	-8.09E1
Probe_3	Probe_3_SVD_0	1.49E-2	1.83E-1	2.24E-1	1.56E-1	-8.76E1

(a) RSM training, error comparison of algorithms; the smaller residuals are given by the DACE algorithm.

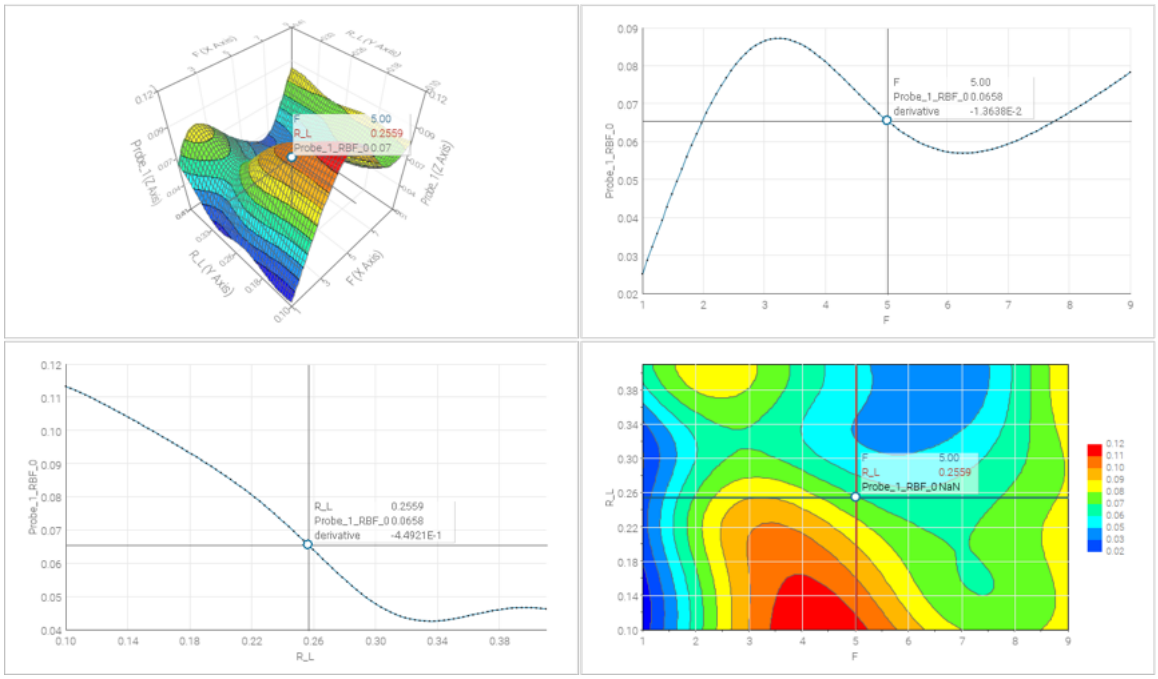


(b) DACE - best fitting

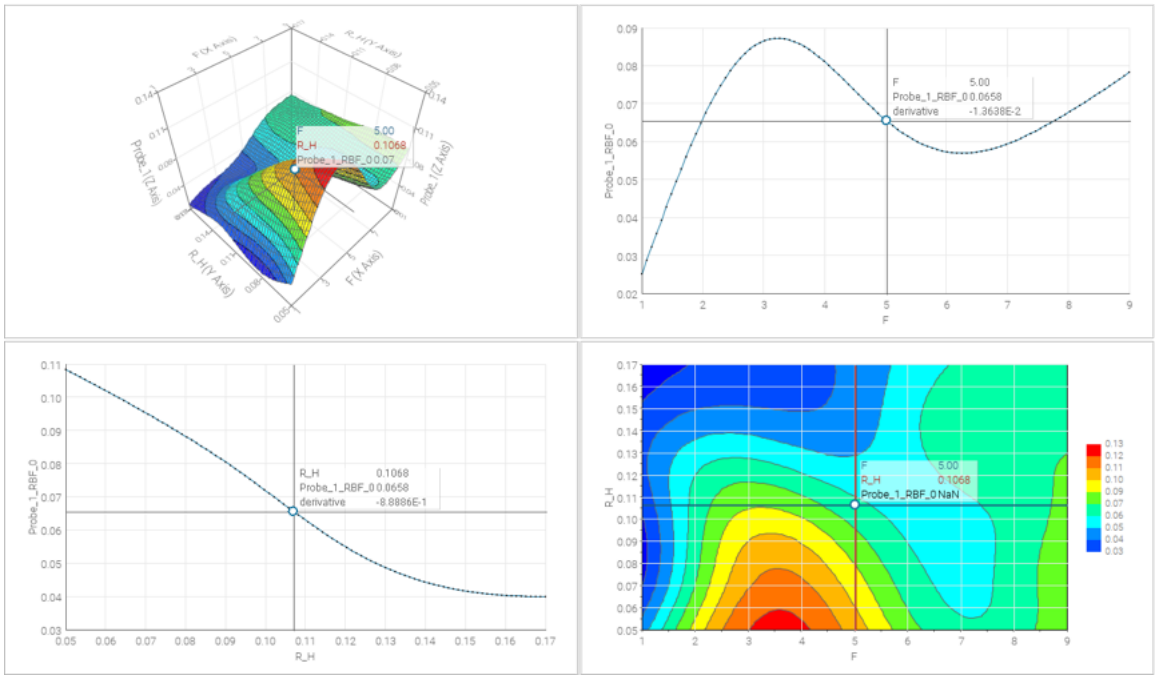


(c) STEP - bad fitting

Figure 5.11: RSM training of Probe 3: (a) error comparison of RSM algorithms; (b) the best fitting algorithm; (c) example of bad fitting algorithm.

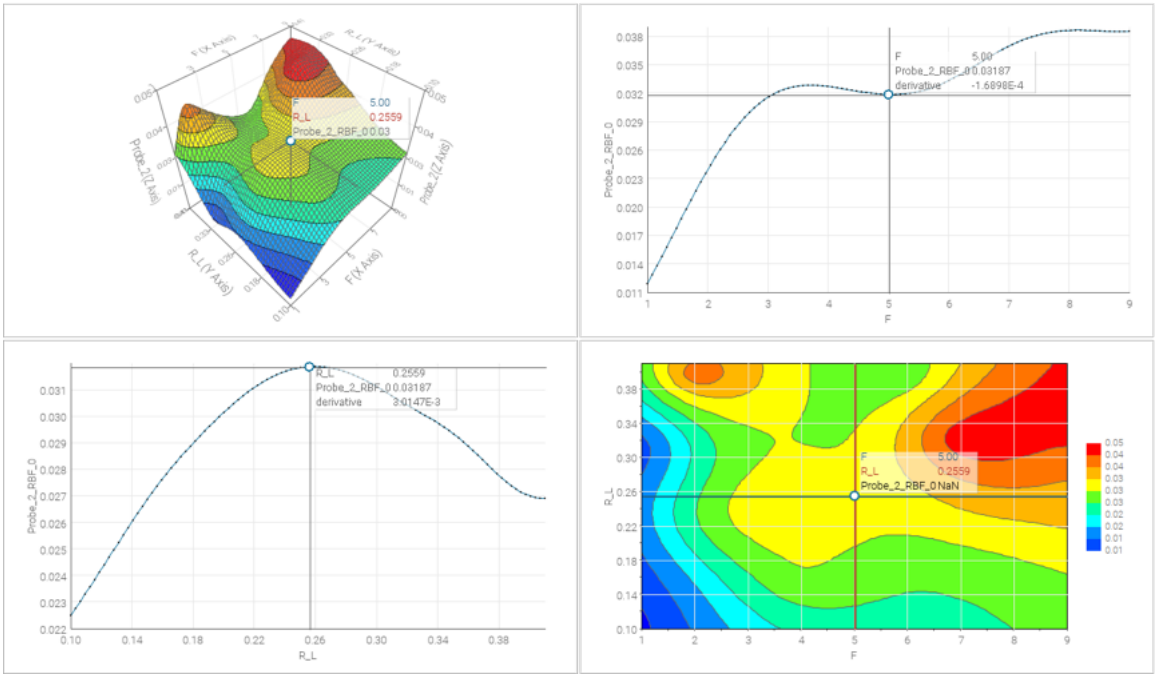


(a) 3D plot of the response model, frequency and R_L behaviour at constant $R_H = 0.1068$ m.

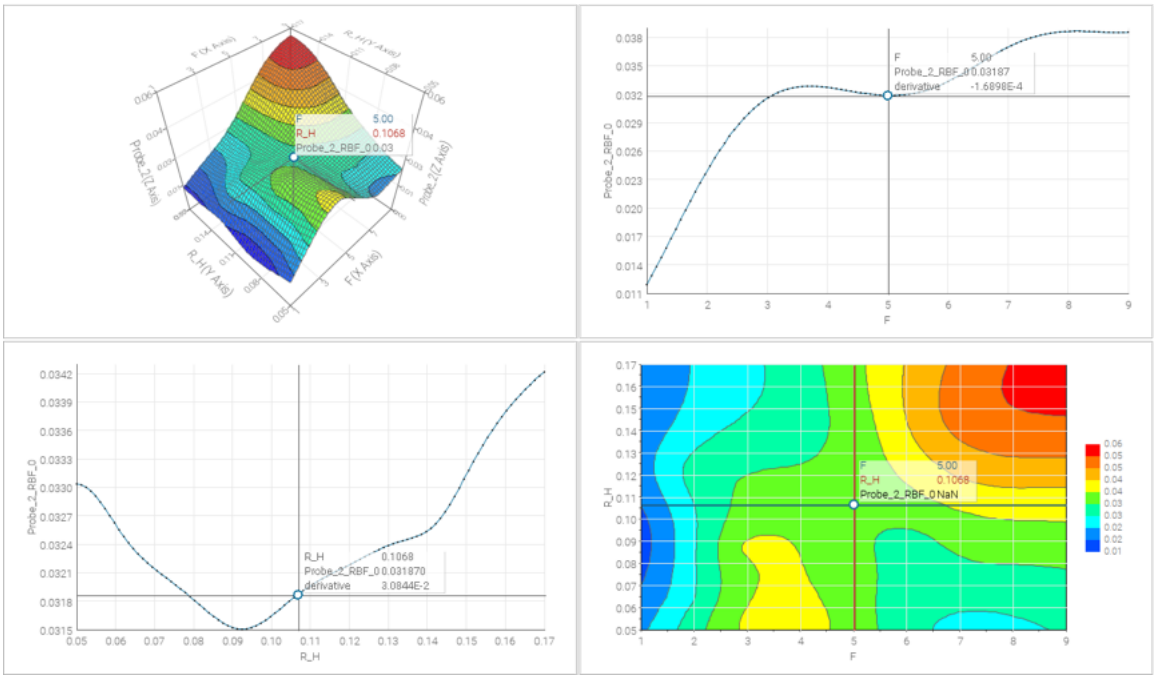


(b) 3D plot of the response model, frequency and R_L behaviour at constant $R_L = 0.2559$ m.

Figure 5.12: 3D plot of the response model for P1 (a) at constant R_H and (b) at constant R_L. Each sub-figure consists of four panels: on the top left side the actual 3D plot; on the top right side 2D plot of the wave height at variable frequency; on the bottom left side 2D plot of the wave height at variable recess measure; on the bottom right side 2D plot of frequency and recess measure correlation, the color indicated the wave height variation.

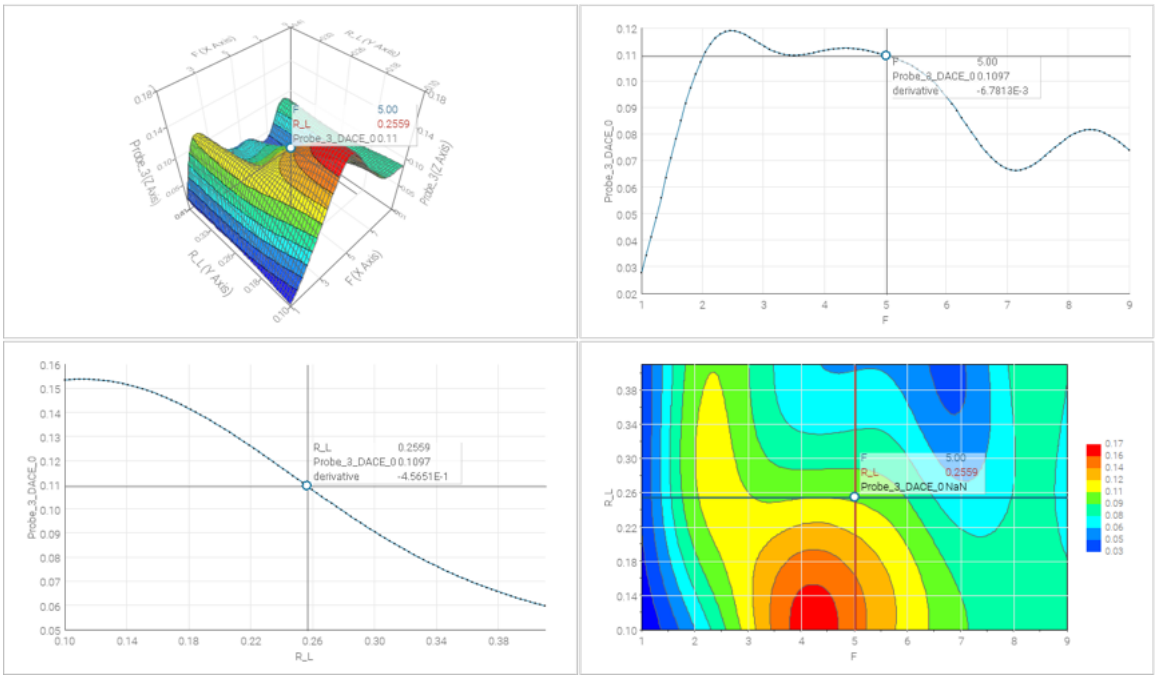


(a) 3D plot of the response model, frequency and R_L behaviour at constant $R_L = 0.2559$ m.

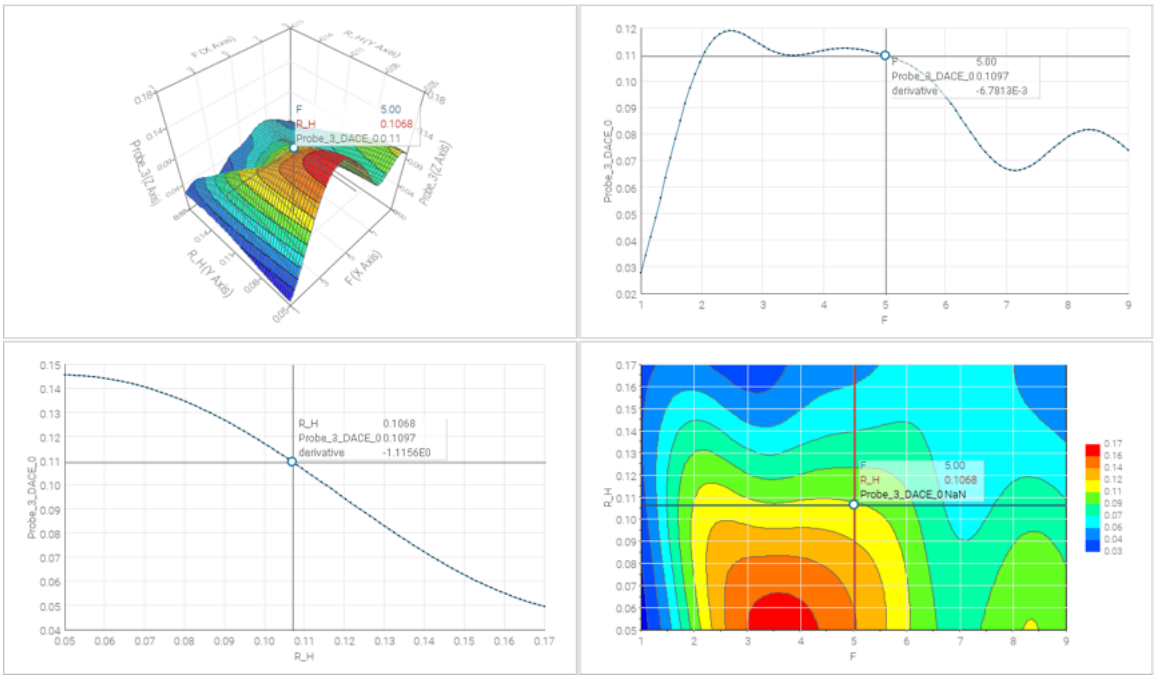


(b) 3D plot of the response model, frequency and R_L behaviour at constant $R_L = 0.2559$ m.

Figure 5.13: 3D plot of the response model for P2 (a) at constant R_H and (b) at constant R_L. Each sub-figure consists of four panels: on the top left side the actual 3D plot; on the top right side 2D plot of the wave height at variable frequency; on the bottom left side 2D plot of the wave height at variable recess measure; on the bottom right side 2D plot of frequency and recess measure correlation, the color indicated the wave height variation.



(a) 3D plot of the response model, frequency and R_L behaviour at constant $R_L = 0.2559$ m.



(b) 3D plot of the response model, frequency and R_L behaviour at constant $R_L = 0.2559$ m.

Figure 5.14: 3D plot of the response model for P3 (a) at constant R_H and (b) at constant R_L. Each sub-figure consists of four panels: on the top left side the actual 3D plot; on the top right side 2D plot of the wave height at variable frequency; on the bottom left side 2D plot of the wave height at variable recess measure; on the bottom right side 2D plot of frequency and recess measure correlation, the color indicated the wave height variation.

5.3.4 The optimisation with the RSM

The preferred algorithms for the RSM are marked as favourites: these algorithms serve as the default choices for calculating the estimation of the new DOE in the Planner environment, which is entirely virtual due to the implementation of the RSM. RSM-based runs are highly efficient, allowing the execution of a substantial number of designs. We seamlessly obtained about 5000 virtual simulations in a few minutes. From these simulations, we select the best ones, specifically those positioned on the Pareto front, that will be further validated by the simulation with the corresponding input values. The Pareto front in Fig. 5.15 represents a common trend when having two conflicting objectives in two-dimensional analysis.

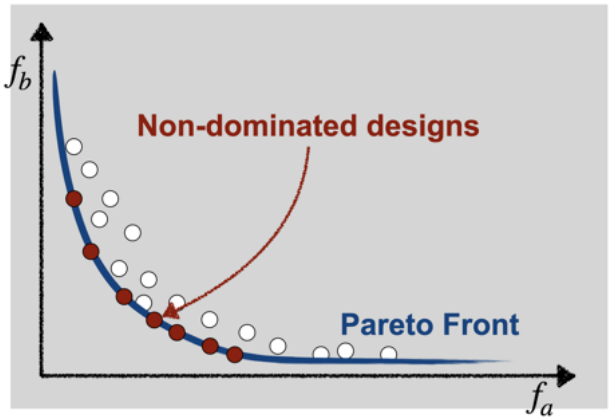
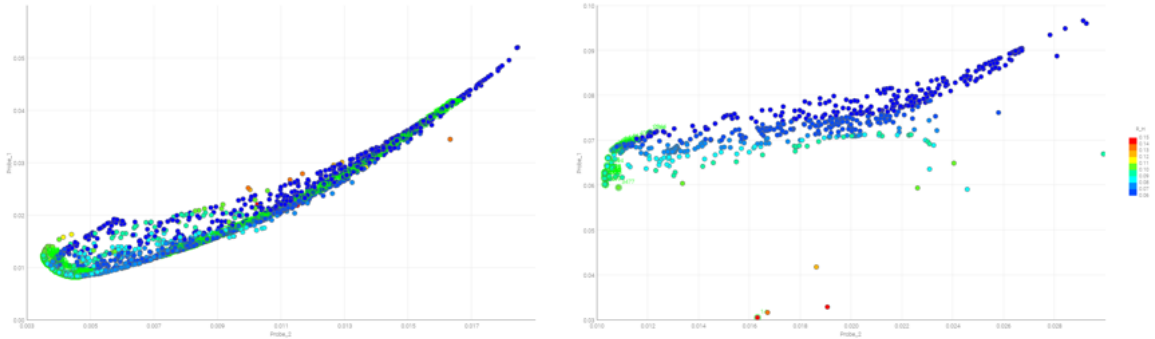


Figure 5.15: Representation of common Pareto front in two dimensional, of two conflicting objectives.

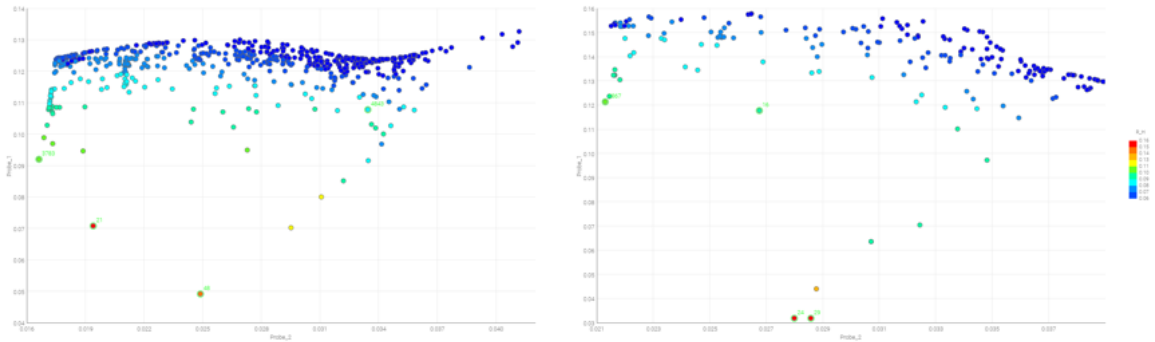
Initially, the virtual designs are distributed into subsets of thousands of samples based on the frequency value. The subsets are depicted in Fig. 5.16; every bubble plot has P1 as abscissa and P2 as ordinate. Then, by applying the Pareto front to each subset, we identify the non-dominated results the in Fig. 5.16 are detected by green tags. The values belonging to the Pareto are not ordered following the trend in Fig. 5.15, which is very common when dealing with multi-objective analysis as the present one. Regrettably, they still amount to a few dozen, necessitating a decision-making process, namely human intervention. Assuming that most critical water behaviour occurs at P3 (confirmed in Fig. 4.9), we prioritise the designs that minimize the wave height at P3; conveniently, we can get few (maximum three) samples per frequency.

5.3.5 Final results and discussion

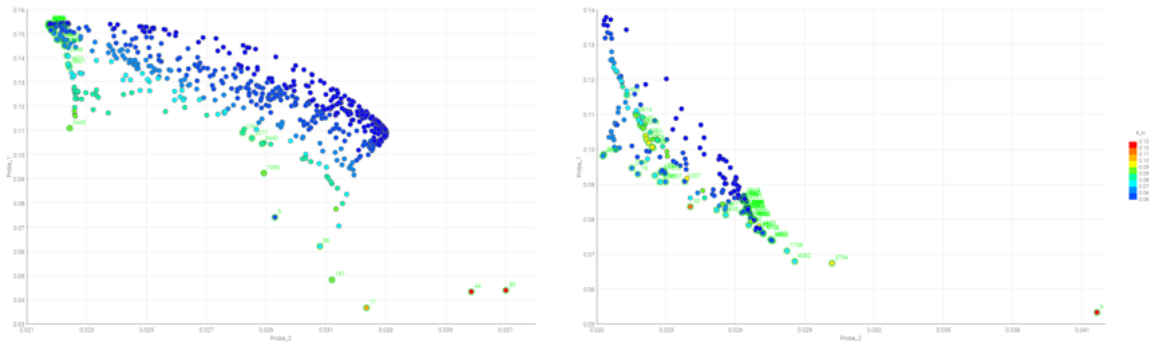
The outcomes obtained virtually are merely predictions. Therefore, it is important to compare them with corresponding simulations to assess their accuracy. We have chosen a threshold for residuals at 15%. This choice seems acceptable, especially considering that the mean errors from



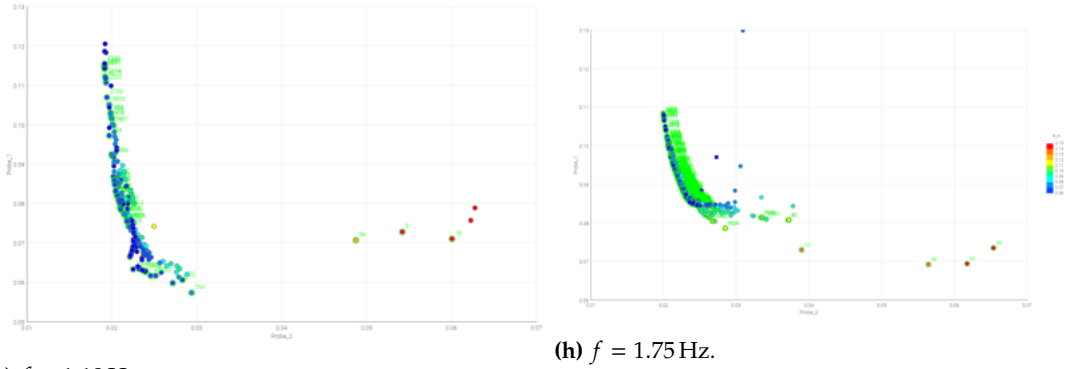
(a) $f = 0.47\,\text{Hz}$ **(b)** $f = 0.71\,\text{Hz}$



(c) $f = 0.79\,\text{Hz}$ **(d)** $f = 0.87\,\text{Hz}$



(e) $f = 0.93\,\text{Hz}$. **(f)** $f = 1.01\,\text{Hz}$.



(g) $f = 1.18\,\text{Hz}$. **(h)** $f = 1.75\,\text{Hz}$.

Figure 5.16: Every bubble plot shows virtual designs obtained with RSM per frequency value in the set $f = [0.47; 0.71; 0.79; 0.87; 0.93; 1.01; 1.18; 1.75]$ Hz. Non-dominated results of Pareto front marked with green tags.

training the response models are typically higher than 5%. The initial set of frequency values was $f = [0.47; 0.71; 0.79; 0.87; 0.93; 1.01; 1.18; 1.75]$ Hz, but only designs of five frequencies have resulted acceptable, precisely the subset $f = [0.87; 0.93; 1.01; 1.18; 1.75]$ Hz. However, we think that increasing the number of runs for the training could result in a better fit for the RSM. In Tab. 5.1 the best resulting designs are presented, their virtual and real wave heights at the probes positions and the value of residuals, meaning the relative error between the virtual and the real results e_r.

Table 5.1: Comparison between RSM virtual designs and simulation results. In the table columns, from left to right: the frequency, recess measurements R_H and R_L, wave height at probes estimated by the RSM, wave height at probes resulted from simulations and e_r, the relative error between these value, in percentage.

			Virtual [m]			Real [m]			e_r [%]		
f [Hz]	R_H [m]	R_L [m]	P1	P2	P3	P1	P2	P3	P1	P2	P3
0.87	0.1527	0.3094	0.0320	0.0286	0.0385	0.0273	0.0254	0.0439	14%	11%	14%
0.93	0.1074	0.2971	0.0482	0.0312	0.0911	0.0508	0.0308	0.0879	5%	1%	4%
1.01	0.0500	0.2707	0.0797	0.0268	0.1060	0.0864	0.0259	0.1182	8%	4%	12%
1.18	0.0707	0.2961	0.0607	0.0283	0.0673	0.0557	0.0313	0.0674	8%	10%	0%
1.75	0.1157	0.2087	0.0808	0.0371	0.0706	0.0767	0.0376	0.0698	5%	1%	1%

The first three columns in Tab. 5.1 refer to the input parameters, frequency, R_H, and R_L, then there are the objective function responses obtained with the RSM, adjacent there are the results given by simulating the corresponding input values and finally the value of e_r. This set of designs has proven to remain below the threshold 15% and therefore acceptable. The designs are depicted in the scatter plot in Fig. 5.17 where the frequency gives the x-coordinate in Hertz and the y-coordinate indicate the recess measurements for the resulted designs. The designs are defined by the values of R_H and R_L, which are plotted respectively with yellow dot markers and green triangle markers. In Fig. 5.17 are collected the geometries obtained for the subset of frequencies $f = [0.87; 0.93; 1.01; 1.18; 1.75]$ Hz.

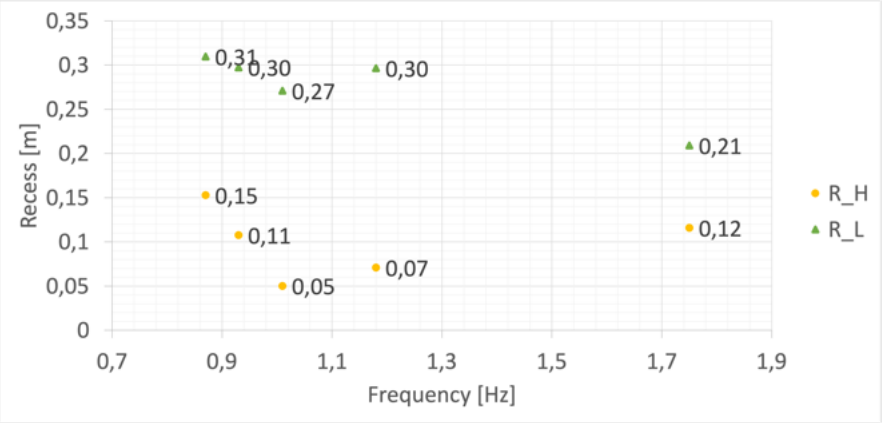
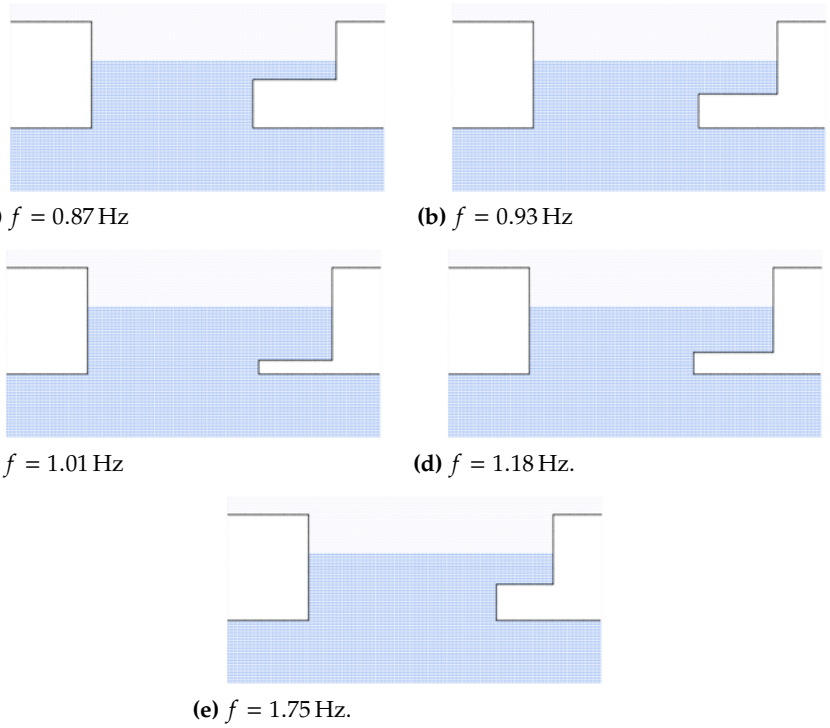


Figure 5.17: Plot of the designs, with frequency [Hz] values in x-coordinate and in y-coordinate the measurements in [m] of the recess designs. R_H and R_L are represented respectively with yellow dot markers and green triangle markers. Optimisation results, comparison between virtual and real designs

The investigation done in this chapter must be regarded as qualitative due the number of approximations taken during its implementation. Nevertheless, in Fig. 5.17, there is a visible trend of the R_H and R_L



(a) $f = 0.87\,\text{Hz}$ **(b)** $f = 0.93\,\text{Hz}$

(c) $f = 1.01\,\text{Hz}$ **(d)** $f = 1.18\,\text{Hz}$.

(e) $f = 1.75\,\text{Hz}$.

Figure 5.18: In (a), (b), (c), (d) and (e) final geometries for the recess in the moon pool based on frequency value $f = [0.87; 0.93; 1.01; 1.18; 1.75]\,\text{Hz}$.

values that depends on the fluid motion, which seems to confirm the presence of local minima for both the values of R_L and R_H, as it was suggested in [81]. We recommend continuing this study in the future, improving the moon pool case modelling, and incrementing the type of wave conditions. However, the current SPH methods, especially within our code, might not be the best viable tool to handle the problem with. The code is not yet ready to handle complex test cases as the present one: not only further developments would be necessary developments to the boundary treatments, including open boundaries, wave makers, and wave absorption conditions, but the simulations should be three-dimensional and therefore considerably more time-consuming.

Conclusions 6

Motivated by the idea that the particle approaches are a viable choice for simulating the interface evolution and the density and viscosity jump between different phases, we have partially explored the field of multiphase flows modelling some selected cases with the Lagrangian SPH method. Nowadays, SPH is an accredited method in the realm of CFD methods, and its applicability to multiphase flows has already been tested. Nonetheless, a variety of complexities must be accounted for when investigating such flows, both for academic and industrial applications, that have not yet been addressed in the literature. The present work focuses mostly on selected application examples of multiphase interfacial flows that require, among others, an adequate SPH approximation for the viscous force.

The manuscript includes a considerable literature review on the state of the art in Chapter 1. Chapter 2 is an insight into the mathematical model that is the foundation of the SPH method; additionally, the governing equations that characterise the current SPH method are described.

In Chapter 3, we have presented the mVER formulation for the viscous force term based on the proposal by Violeau [192]. In Violeau's formula, the inter-particle dynamic viscosity is found by implementing the arithmetic mean of the viscosity coefficients of the phases. In the mVER force, the inter-particle dynamic viscosity is given by the harmonic mean of the viscosities; this modification was inspired by [77]. The validity of this proposal was proven by testing the equation within the benchmark of the Poiseuille flow in single- and two-phase scenarios. We have compared the analytical velocity profile of the flows (both scenarios) with SPH results computed using three formulations: the Hu and Adams one, the Violeau one, and the mVER one. The velocity profile in the first case, single-phase Poiseuille flow, was performed efficiently by all three formulations. Regarding the second case, only the Hu and Adams and the mVER formulations have matched the benchmark. The Violeau formulation has shown a too-viscous behaviour due to the implementation of the arithmetic mean. Therefore, the harmonic mean used in the mVER formula has been confirmed to be a better choice. The results obtained with the mVER and the Hu and Adams formulations are very close, and the research on their differences is still an open discussion for future work.

The mVER has demonstrated its efficacy by accurately computing the benchmarks of the rising bubble by [91], which features two regimes of the bubbles that differ on the type of shape that the bubble takes due to buoyancy and the shear effect of the external fluid. In the first case, the density and viscosity ratios between the two fluid phases are equal to

10; the regime is called ellipsoidal because the final shape of the bubble is ellipsoidal. The SPH bubble at the end of the simulation has a shape comparable to the reference benchmark. In the second case, the density and viscosity ratios are much higher than in the first regime, and are 1000 and 100, respectively. These new conditions lead the bubble to a more deformed shape; for this regime, it is expected to have the so-called skirt under the bubble's main body. The second regime has proved to be more challenging with respect to the first one, and despite the main body of the bubble matching the benchmark, we could not reproduce the elongated skirt. Possibly, a considerable increase in the resolution (thus increasing the number of particles) would allow the formation of the skirt, but the machine power requested could be extreme (the SPH simulation of the skirted regime presented in this manuscript lasted about a week). Both bubble scenarios present visible oscillations in the rising velocity. This is a well-known issue that occurs in weakly compressible approaches, including WCSPH: even small oscillations in density can be visible in other quantities that depend on it, such as pressure and velocity. However, this problem affects the accuracy of the results, and deeper investigations are needed to understand the reasons and mitigate the effects. In this work, we have partially tested the influence of the kernel, implementing the rising bubble within the quintic spline function, which nonetheless produces an oscillating outcome.

The proposed viscous force formulation was also validated in Chapter 4 against the sloshing tank benchmark, a study case presented by [75]. The referenced paper simulates a thin layer of inviscid fluid with water density in a 2D tank; the free-surface flow is excited by a horizontal force that is time-periodic. We have reproduced this test in our SPH code. However, the flow is multiphase: the two viscous phases have air-water properties. The SPH outcomes for wave height and sloshing force at the wall are verified against data obtained with the analytical formula for potential flow by [61]. Both the quantities have shown good agreement with the analytical: the SPH wave height has almost the same magnitude and period as the theoretical one; the sloshing force meets the periodic trend, too, whereas the magnitude is smaller due to the effect of viscosity. The sloshing tank has been a natural study to initiate a more complex work, the SPH modelling of the moon pool, which, to the best of the author's knowledge, has not been previously tested using SPH. The moon pool is an opening through the hull that features some kind of vessel; it allows for underwater activities. Our work on the moon pool is inspired by the scaled model by [79]; in this study are presented experimental data and 3D numerical results of the flow behaviour inside the moon pool of a captive vessel. For practical reasons, the model simulated in our code: (i) consists only of the part of the vessel that includes the moon pool; (ii) is 2D; (iii) is multiphase and the fluid phases have air and water properties; (iv) is characterised by a flow excited by a periodic sloshing force; (v) is horizontally bounded by period boundary conditions. The simulations of our moon pool model have proved to be stable, and the

frequency of the sloshing flow inside the moon pool is comparable with the benchmark.

The moon pool we have been focused on has a recess, namely a step that gives the characteristic L-shape geometry. Although the recess is helpful for some activities, its presence affects the flow inside the moon pool; in fact, it acts as an artificial local sea bottom, boosting the effect of sloshing.

In collaboration with ESTECO, we have coupled the SPH code with their software modeFRONTIER. We have utilised the latter to optimise the geometry of the recess (its length and height), aiming to minimise the sloshing wave at specific locations in the moon pool (left wall, center and right wall) for variable conditions of the flow. We have performed a thorough statistical analysis, employing the RSM + DOE optimisation technique recommended by the ESTECO team. The first step of this type of optimisation is to produce the DOE, a set of simulations in which the input parameters (the recess length and height and the frequency of the external sloshing force) are systematically varied. The precision of the optimisation relies on the number of DOE simulations that can be run. In the moon pool case, the trade-off amounts to just a few dozen runs due to computational constraints. The limited number of simulations run are used to build the response model. This meta-model can predict virtual outcomes (the height of the sloshing wave) varying the geometry of a large number of virtual recess designs. The virtual geometries of the recess obtained with the RSM that formed the Pareto front have been considered the best ones and finally compared with real ones (from simulations). This final test has allowed us to decide whether the virtual geometries predicted by the RSM were accurate. Despite the fact that a few of these final geometries have been discarded, the majority of them have been accepted. The trend of our results is comparable to that reported in the study [81]: for a moderate wavy sea such as the one tested, the sloshing effect is mitigated for a recess length that is about one-third of the moon pool width. However, we consider this study as qualitative due to the necessary simplifications of the physical model implemented. Further investigation with enhanced models of the moon pool (possibly encompassing open boundary conditions) could confirm what has been investigated so far.

In conclusion, while SPH demonstrates the considerable potential for analysing multiphase flows, as evidenced by the promising performance of simulations incorporating the new viscous force, it is not yet sufficiently refined to address the complexities inherent in industrial case studies. Anyway, there is still hope for future progress, so it is important to keep the research going.

Bibliography

[1] Adami, S., Hu, X. Y., and Adams, N.A. 'A generalized wall boundary condition for smoothed particle hydrodynamics'. In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075. DOI: 10.1016/j.jcp.2012.05.005.

[2] Adami, S., Hu, X. Y., and Adams, N.A. 'A transport-velocity formulation for smoothed particle hydrodynamics'. In: *Journal of Computational Physics* 241 (2013), pp. 292–307. DOI: 10.1016/j.jcp.2013.01.043.

[3] Antuono, M. *Stability, accuracy and time-integration issues in SPH schemes*. The Mathemarics of SPH, TU Delft, 24 January2019.

[4] Antuono, M. et al. 'Free-surface flows solved by means of SPH schemes with numerical diffusive terms'. In: *Computer Physics Communications* 181.3 (2010), pp. 532–549. DOI: 10.1016/j.cpc.2009.11.002.

[5] Arai, E. et al. 'Simulating bubble dynamics in a buoyant system'. In: *International Journal for Numerical Methods in Fluids* 92.3 (2020), pp. 169–188. DOI: 10.1002/fld.4778.

[6] Arora, J. S. 'Chapter 10 - Numerical Methods for Unconstrained Optimum Design'. In: *Introduction to Optimum Design (Third Edition)*. Ed. by Jasbir S. Arora. Third Edition. Boston: Academic Press, 2012, pp. 411–441. DOI: https://doi.org/10.1016/B978-0-12-381375-6.00010-3.

[7] Balaban, M. 'Review of Dace-Kriging Metamodel'. In: *Interdisciplinary Description of Complex Systems* 21.3 (2023), pp. 316–323. DOI: 10.7906/indecs.21.3.8.

[8] Bandyopadhyay, S. *Introduction to CUDA programming*. 2023. URL: https://medium.com/@sumitb2015/introduction-to-cuda-programming-c0242f252b5 (visited on 2024).

[9] Batchelor, G. K. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press, 2000. DOI: https://doi.org/10.1017/CBO9780511800955.

[10] Benz, W. 'Applications of Smooth Particle Hydrodynamics (SPH) to astrophysical problems'. In: *Computer Physics Communications* 48.1 (1988), pp. 97–105. DOI: https://doi.org/10.1016/0010-4655(88)90027-6.

[11] Benz, W. and Asphaug, E. 'Explicit 3D Continuum Fracture Modeling with Smooth Particle Hydrodynamics'. In: *Lunar and Planetary Science Conference*. 1993, p. 99.

[12] Bird, B.R., Stewart, W.E., and Lightfoot, E.N. *Transport Phenomena*. Ed. by Wayne Anderson. 2nd ed. John Wiley & Sons, Inc., 1924.

[13] Blatman, G., Sudret, B., and Berveiller, M. 'Quasi random numbers in stochastic finite element analysis'. In: *Mécanique & Industries* 8.3 (2007), pp. 289–297. DOI: 10.1051/meca:2007051.

[14] Bloom, R. and Sekhon, R. *Applied Finite Mathematics*. Ed. by Roberta Bloom. Third. Cupertino, CA.: Mathematics Department, De Anza College, 2016.

[15] Blum, A., Hopcroft, J., and Kannan, R. *Foundations of Data Science*. Cambridge University Press, 2020. DOI: 10.1017/9781108755528.

[16] Bonet, J. and Kulasegaram, S. 'Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations'. In: *International Journal for Numerical Methods in Engineering* 47.6 (2000), pp. 1189–1214. DOI: https://doi.org/10.1002/(SICI)1097-0207(20000228)47:6<1189::AID-NME830>3.0.CO;2-I.

[17] Bourbon. *The moonpool: a safety asset*. 2021. URL: https://www.bourbonoffshore.com/en/news/moonpool-safety-asset (visited on 2024).

[18] Box, G. E. P. and Wilson, K. B. 'On the Experimental Attainment of Optimum Conditions'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 13.1 (1951), pp. 1–38. DOI: https://doi.org/10.1111/j.2517-6161.1951.tb00067.x.

[19] Brackbill, J.U., Kothe, D.B., and Zemach, C. 'A continuum method for modeling surface tension*'. In: *Journal of Computational Physics* 100 (1992).

[20] Brennen, C.E. *Fundamentals of multiphase flow*. 2013, pp. 1–345. DOI: 10.1017/CBO9780511807169.

[21] Britannica, Encyclopedia. *Vilfredo Pareto, Italian economist and sociologist*. 2024. URL: https://www.britannica.com/biography/Vilfredo-Pareto (visited on 2024).

[22] Bush, J. W. M. 'Surface tension'. In: *New Trends in the Physics and Mechanics of Biological Systems: Lecture Notes of the Les Houches Summer School: Volume 92, July 2009*. Oxford University Press, 2011. DOI: 10.1093/acprof:oso/9780199605835.003.0002.

[23] Cahn, J. W. 'Free Energy of a Nonuniform System. II. Thermodynamic Basis'. In: *The Journal of Chemical Physics* 30.5 (2004), pp. 1121–1124. DOI: 10.1063/1.1730145.

[24] Cahn, J.W. and Hilliard, J.E. 'Free energy of a nonuniform system. I. Interfacial free energy'. In: *The Journal of Chemical Physics* 28.2 (1958), pp. 258–267. DOI: 10.1063/1.1744102.

[25] Cahn, J.W. and Hilliard, J.E. 'Free Energy of a Nonuniform System. III. Nucleation in a Two-Component Incompressible Fluid'. In: *The Journal of Chemical Physics* 31.3 (2004), pp. 688–699. DOI: 10.1063/1.1730447.

[26] Chen, J. K., Beraun, J. E., and Carney, T. C. 'A corrective smoothed particle method for boundary value problems in heat conduction'. In: *International Journal for Numerical Methods in Engineering* 46.2 (1999), pp. 231–252. DOI: 10.1002/(SICI)1097-0207(19990920)46:2<231::AID-NME672>3.0.CO;2-K.

[27] Chen, S. and Doolen, G.D. 'Lattice Boltzmann method for fluid flows'. In: *Annual Review of Fluid Mechanics* 30.1 (1998), pp. 329–364. DOI: 10.1146/annurev.fluid.30.1.329.

[28] Chicone, C. 'Chapter 16 - Numerical Methods for Computational Fluid Dynamics'. In: *An Invitation to Applied Mathematics*. Ed. by Carmen Chicone. Academic Press, 2017, pp. 403–510. DOI: https://doi.org/10.1016/B978-0-12-804153-6.50016-6.

[29] Choi, Y. et al. 'Comparison of Factorial and Latin Hypercube Sampling Designs for Meta-Models of Building Heating and Cooling Loads'. In: *Energies* 14.2 (2021). DOI: 10.3390/en14020512.

[30] Clarich, A., Rigoni, E., and Poloni, C. 'A new Algorithm based on Game Theory for Robust and Fast Multi-Objective Optimisation'. In: *Esteco modeFrontier User Guide* (2004).

[31] Cleary, P. W. 'Modelling confined multi-material heat and mass flows using SPH'. In: *Applied Mathematical Modelling* 22 (1998), pp. 981–993. DOI: 10.1016/S0307-904X(98)10031-8.

[32] Clift, R., Grace, J.R., and Weber, M.E. *Bubbles, Drops, and Particles*. ACADEMIC PRESS, INC. (LONDON) LTD., 1978, pp. 201–208.

[33] Colaço, M. J. and Dulikravich, G. S. 'A Survey of Basic Deterministic, Heuristic, and Hybrid Methods for Single-Objective Optimization and Response Surface Generation'. In: *Thermal Measurements and Inverse Techniques*. CRC Press, 2011, pp. 369–420. DOI: 10.1201/b10918-15.

[34] Colagrossi, A., Antuono, M., and Le Touzé, D. 'Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model'. In: *Physical Review E* 79.5 (2009), p. 056701. DOI: 10.1103/PhysRevE.79.056701.

[35] Colagrossi, A. and Landrini, M. 'Numerical simulation of interfacial flows by smoothed particle hydrodynamics'. In: *Journal of Computational Physics* 191.2 (2003), pp. 448–475. DOI: 10.1016/S0021-9991(03)00324-3.

[36] Colagrossi, A. et al. 'A study of violent sloshing wave impacts using an improved SPH method'. In: *Journal of Hydraulic Research* 48.sup1 (2010), pp. 94–104. DOI: 10.1080/00221686.2010.9641250.

[37] Colagrossi, A. et al. 'Discussion of Stokes' hypothesis through the smoothed particle hydrodynamics model'. In: *Physical Review E* 96.2 (2017), p. 023101. DOI: 10.1103/PhysRevE.96.023101.

[38] Colagrossi, A. et al. 'Reynolds number and Shallow Depth Sloshing'. In: *3rd ERCOFTAC SPHERIC Workshop on SPH Applications* December 2016 (2008), pp. 221–228.

[39] Courant, R., Friedrichs, K., and Lewy, H. 'Über die partiellen Differenzengleichungen der mathematischen Physik'. In: *Mathematische Annalen* 100 (1928), pp. 32–74.
DOI: 10.1007/BF01448839.

[40] Crespo, A. J. C. 'Application of the Smoothed Particle Hydrodynamics model SPHysics to free-surface hydrodynamics'. PhD thesis. Universitade de Vigo Departemento de fisica aplicada PhD, 2008.

[41] Crespo, A. J. C. et al. 'High Performance Of SPH codes: Best approaches for efficient parallelisation on GPU computing'. In: *4th International SPHERIC Workshop* (2009), pp. 69–76.

[42] Cummins, S. J. and Rudman, M. 'An SPH Projection Method'. In: *Journal of Computational Physics* 152.2 (1999), pp. 584–607. DOI: 10.1006/jcph.1999.6246.

[43] Dalrymple, R. A. and Knio, O. 'SPH modelling of water waves'. English (US). In: *Coastal Dynamics 2001*. Coastal Dynamics 2001. 4th Conference on Coastal Dynamics ; Conference date: 11-06-2001 Through 15-06-2001. United States: American Society of Civil Engineers (ASCE), 2001, pp. 779–787.
DOI: 10.1061/40566(260)80.

[44] Das, A.K. and Das, P.K. 'Bubble evolution through submerged orifice using smoothed particle hydrodynamics: Basic formulation and model validation'. In: *Chemical Engineering Science* 64.10 (2009), pp. 2281–2290.
DOI: https://doi.org/10.1016/j.ces.2009.01.053.

[45] Dehnen, W. and Aly, H. 'Improving convergence in smoothed particle hydrodynamics simulations without pairing instability'. In: *Monthly Notices of the Royal Astronomical Society* 425.2 (2012), pp. 1068–1082.
DOI: 10.1111/j.1365-2966.2012.21439.x.

[46] Di Mascio, A. et al. 'Smoothed Particle Hydrodynamics method from a large eddy simulation perspective'. In: *Physics of Fluids* 29.3 (2017), p. 035102. DOI: 10.1063/1.4978274.

[47] Direct, Science. *Artificial Viscosity*. 2016. URL: https://www.sciencedirect.com/topics/mathematics/artificial-viscosity (visited on 2024).

[48] Direct, Science. *Central Processing Unit*. 2017. URL: https://www.sciencedirect.com/topics/engineering/central-processing-unit (visited on 2024).

[49] Direct, Science. *Graphics Processing Unit*. 2023. URL: https://www.sciencedirect.com/topics/engineering/graphics-processing-unit (visited on 2024).

[50] Domínguez, J. M., Crespo, A. J. C., and Gómez-Gesteira, M. 'Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method'. In: *Computer Physics Communications* 184.3 (2013), pp. 617–627. DOI: 10.1016/j.cpc.2012.10.015.

[51] Domínguez, J. M. et al. 'New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters'. In: *Computer Physics Communications* 184.8 (2013), pp. 1848–1860. DOI: 10.1016/j.cpc.2013.03.008.

[52] Duan, G., Yamaji, A., and Sakai, M. 'An incompressible–compressible Lagrangian particle method for bubble flows with a sharp density jump and boiling phase change'. In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113425. DOI: 10.1016/j.cma.2020.113425.

[53] Duan, G. et al. 'Stable multiphase moving particle semi-implicit method for incompressible interfacial flow'. In: *Computer Methods in Applied Mechanics and Engineering* 318 (2017), pp. 636–666. DOI: 10.1016/j.cma.2017.01.002.

[54] Durakovic, B. 'Design of experiments application, concepts, examples: State of the art'. In: *Periodicals of Engineering and Natural Sciences (PEN)* 5.3 (2017), pp. 421–439. DOI: 10.21533/pen.v5i3.145.

[55] Dwornicka, R. and Pietraszek, J. 'The outline of the expert system for the design of experiment'. In: *Production Engineering Archives* 20.20 (2018), pp. 43–48. DOI: 10.30657/pea.2018.20.09.

[56] Dynamic Positioning Committee, J. Steinbeck. *Hisotry of DP*. 2007. URL: https://dynamic-positioning.com/history-of-dp/ (visited on 2024).

[57] Encyclopedia Britannica, B. H. Caudle and McLeroy, P. G. *Deep and ultradeep water*. 2017. URL: https://www.britannica.com/technology/petroleum-production/Deep-and-ultradeep-water (visited on 2024).

[58] Español, P. and Revenga, M. 'Smoothed dissipative particle dynamics'. In: *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* 67.2 (2003), p. 12. DOI: 10.1103/PhysRevE.67.026705.

[59] Evrard, A.E. 'Beyond N-body: 3D cosmological gas dynamics'. In: *Monthly Notices of the Royal Astronomical Society* 235.3 (1988), pp. 911–934. DOI: 10.1093/mnras/235.3.911.

[60] Faltinsen, O. M., Rognebakke, O. F., and N., Timokha A. 'Two-dimensional resonant piston-like sloshing in a moonpool'. In: *Journal of Fluid Mechanics* 575 (2007), pp. 359–397. DOI: 10.1017/S002211200600440X.

[61] Faltinsen, O.M. and Timokha, A.N. *Sloshing*. Cambridge University Press, 2009.

[62] Fedkiw, R. P. et al. 'A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)'. In: *Journal of Computational Physics* 152.2 (1999), pp. 457–492. DOI: https://doi.org/10.1006/jcph.1999.6236.

[63] Fernandez-Granda, C. *The Singular-Value Decomposition*. 2019. URL: https://cims.nyu.edu/~cfgranda/pages/MTDS_spring19/notes/svd.pdf (visited on 2024).

[64] Ferrand, M. et al. 'Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method'. In: *International Journal for Numerical Methods in Fluids* 71.4 (2013), pp. 446–472. DOI: https://doi.org/10.1002/fld.3666.

[65] Filho, C. A. *Smoothed Particle Hydrodynamics: Fundamentals and Basic Applications in Continuum Mechanics*. Springer International Publishing, 2018.

[66] Fisher, R. A. 'Studies in crop variation. I. An examination of the yield of dressed grain from Broadbalk'. In: *The Journal of Agricultural Science* 11.2 (1921), pp. 107–135. DOI: 10.1017/S0021859600003750.

[67] Fisher, R. A. *The Design of Experiments*. 1st ed. Vol. 137. Edinburgh: Oliver & Boyd Press, 1935.

[68] Fukuda, K. 'Behavior of Water in Vertical Well with Bottom Opening of Ship, and its Effects on Ship-Motion'. In: *Journal of the Society of Naval Architects of Japan* 1977 (1977), pp. 107–122.

[69] G., Pradeep. *CUDA refresher: the CUDA programming model*. 2020. URL: https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/ (visited on 2024).

[70] Gaillarde, G. and A., Cotteleer. 'Water Motion in Moonpools, Empirical and Theoretical Approach'. In: *Maritime Research Institute Nether- lands* 2435 (2004).

[71] 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 (1977), pp. 375–389. DOI: 10.1093/mnras/181.3.375.

[72] Glimm, J. 'Tracking of Interfaces for Fluid Flow: Accurate Methods for Piecewise Smooth Problems'. In: *Transonic, Shock, and Multidimensional Flows*. Ed. by R. E. Meyer. Academic Press, 1982, pp. 259–287. DOI: https://doi.org/10.1016/B978-0-12-493280-7.50016-4.

[73] Gotoh, H. and Khayyer, A. 'On the state-of-the-art of particle methods for coastal and ocean engineering'. In: *Coastal Engineering Journal* 60.1 (2018), pp. 79–103. DOI: 10.1080/21664250.2018.1436243.

[74] Green, M. D. 'Sloshing simulations with the smoothed particle hydrodynamics (SPH) method'. PhD thesis. Imperial College London, 2017.

[75] Green, M. D. and Peiró, J. 'Long duration SPH simulations of sloshing in tanks with a low fill ratio and high stretching'. In: *Computers & Fluids* 174 (2018), pp. 179–199. DOI: 10.1016/j.compfluid.2018.07.006.

[76] Grenier, N. et al. 'An Hamiltonian interface SPH formulation for multi-fluid and free surface flows'. In: *Journal of Computational Physics* 228.22 (2009), pp. 8380–8393. DOI: 10.1016/j.jcp.2009.08.009.

[77] Grenier, N. et al. 'Viscous bubbly flows simulation with an interface SPH model'. In: *Ocean Engineering* 69 (2013), pp. 88–102. DOI: 10.1016/j.oceaneng.2013.05.010.

[78] Grucelski, A. and Pozorski, J. 'Lattice Boltzmann simulations of flow past a circular cylinder and in simple porous media'. In: *Computers & Fluids* 71 (2013), pp. 406–416. DOI: https://doi.org/10.1016/j.compfluid.2012.11.006.

[79] Guo, X. et al. 'Resonant water motions within a recessing type moonpool in a drilling vessel'. In: *Ocean Engineering* 129.January (2017), pp. 228–239. DOI: 10.1016/j.oceaneng.2016.11.030.

[80] Hamill, P. *A Student's Guide to Lagrangians and Hamiltonians*. Cambridge University Press, 2013. DOI: 10.1017/CBO9781107337572.

[81] Hammargren, E. and Törnblom, J. 'Effect of the Moonpool on the Total Resistance of a Drillship'. MA thesis. Chalmers University of Technology, 2012, pp. 1–2.

[82] Hammersley, J. M. and Handscomb, D. C. *Monte Carlo Methods*. 1. Dordrecht: Springer Netherlands, 1964, pp. 1–14. DOI: 10.1007/978-94-009-5819-7.

[83] Hardy, R. L. 'Multiquadric equations of topography and other irregular surfaces'. In: *Journal of Geophysical Research (1896-1977)* 76.8 (1971), pp. 1905–1915. DOI: https://doi.org/10.1029/JB076i008p01905.

[84] Harlow, F. H. and Welch, J. E. 'Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface'. In: *Physics of Fluids* 8.12 (1965), p. 2182. DOI: 10.1063/1.1761178.

[85] Haykin, S. S. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999.

[86] Hirt, C. W., Cook, J. L., and Butler, T. D. 'A lagrangian method for calculating the dynamics of an incompressible fluid with free surface'. In: *Journal of Computational Physics* 5.1 (1970), pp. 103–124. DOI: 10.1016/0021-9991(70)90055-0.

[87] Hu, X. Y. and Adams, N. A. 'A multi-phase SPH method for macroscopic and mesoscopic flows'. In: *Journal of Computational Physics* 213.2 (2006), pp. 844–861. DOI: 10.1016/j.jcp.2005.09.001.

[88] Hu, X. Y. and Adams, N. A. 'Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows'. In: *Physics of Fluids* 18.10 (2006), p. 101702. DOI: 10.1063/1.2359741.

[89] Huang, X. H. et al. 'An experimental investigation of reduction effect of damping devices in the rectangular moonpool'. In: *Ocean Engineering* 196.October (2020), p. 106767. DOI: 10.1016/j.oceaneng.2019.106767.

[90] HUT / Aalto. *History of Optimization*. URL: http://www.mitrikitti.fi/opthist.html (visited on 2024).

[91] Hysing, S. et al. 'Quantitative benchmark computations of two-dimensional bubble dynamics'. In: *International Journal for Numerical Methods in Fluids* 60.11 (2009), pp. 1259–1288. DOI: 10.1002/fld.1934.

[92] Kajzer, A. 'Modelling of interfacial flows with the diffuse-interface method using weakly compressible approach'. PhD thesis. Institute of Fluid-Flow Machinery, Polish Academy of Sciences, 2020.

[93] Kajzer, A. and Pozorski, J. 'Application of the Entropically Damped Artificial Compressibility model to direct numerical simulation of turbulent channel flow'. In: *Computers & Mathematics with Applications* 76.5 (2018), pp. 997–1013. DOI: 10.1016/j.camwa.2018.05.036.

[94] Kajzer, A., Pozorski, J., and Szewc, K. 'Large-eddy simulations of 3D Taylor-Green vortex: comparison of Smoothed Particle Hydrodynamics, Lattice Boltzmann and Finite Volume methods'. In: *Journal of Physics: Conference Series* 530.1 (2014), p. 012019. DOI: 10.1088/1742-6596/530/1/012019.

[95] Koshizuka, S. and Oka, Y. 'Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid'. In: *Nuclear Science and Engineering* 123.3 (1996), pp. 421–434. DOI: 10.13182/NSE96-A24205.

[96] Krimi, A., Jandaghian, M., and Shakibaeinia, A. 'A WCSPH Particle Shifting Strategy for Simulating Violent Free Surface Flows'. In: *Water* 12.11 (2020), p. 3189. DOI: 10.3390/w12113189.

[97] Krimi, A. et al. 'Smoothed Particle Hydrodynamics: A consistent model for interfacial multiphase fluid flow simulations'. In: *Journal of Computational Physics* 358 (2018), pp. 53–87. DOI: 10.1016/j.jcp.2017.12.006.

[98] Kumar, A., Pandit, B., and Tiwari, G. 'Reliability-based stability analysis of large rock slopes with different failure mechanisms using response surface methodology'. In: *Environmental Earth Sciences* 81.21 (2022), pp. 1–23. DOI: 10.1007/s12665-022-10624-1.

[99] Landau, L. D. and Lifshitz, E. M. *Fluid Mechanics*. Elsevier, 1987. DOI: 10.1016/C2013-0-03799-1.

[100] Landrini, M., Colagrossi, A., and Faltinsen, O. 'Sloshing in 2D Flows by the SPH Method'. In: *In 8th Int. Conf. no Num. Ship Hydrodynamics. Busan. Korea (Sept 2003), pp. 1-15* August (2003), pp. 1–15.

[101] Landrini, M. et al. 'Gridless simulations of splashing processes and near-shore bore propagation'. In: *Journal of Fluid Mechanics* 591.June (2007), pp. 183–213. DOI: 10.1017/S0022112007008142.

[102] Lawson, J. *Design and Analysis of Experiments with R*. Raton, USA: CRC Press Taylor & Francis Group, 2015.

[103] 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 Harold E. Trease, Martin F. Fritts, and W. Patrick Crowley. Vol. 395. 1991, pp. 248–257. DOI: 10.1007/3-540-54960-9_58.

[104] Libersky, L. D. et al. 'High Strain Lagrangian Hydrodynamics: A Three-Dimensional SPH Code for Dynamic Material Response'. In: *Journal of Computational Physics* 109.1 (1993), pp. 67–75. DOI: https://doi.org/10.1006/jcph.1993.1199.

[105] Lin, C. D. and Tang, B. 'Latin Hypercubes and Space-filling Designs'. In: *Handbook of Design and Analysis of Experiments*. New York: Chapman and Hall/CRC, 2014. Chap. Chapter 17, pp. 613–646. DOI: 10.48550/arXiv.2203.06334.

[106] Lind, S. J., Rogers, B. D., and Stansby, P. K. 'Review of smoothed particle hydrodynamics: Towards converged Lagrangian flow modelling: Smoothed Particle Hydrodynamics review'. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2241 (2020). DOI: 10.1098/rspa.2019.0801.

[107] Liu, G. R. and Liu, M. B. *Smoothed Particle Hydrodynamics*. World Scientif, 2003. DOI: 10.1142/5340.

[108] Liu, M. B., Liu, G. R., and Lam, K. Y. 'Constructing smoothing functions in smoothed particle hydrodynamics with applications'. In: *Journal of Computational and Applied Mathematics* 155.2 (2003), pp. 263–284. DOI: 10.1016/S0377-0427(02)00869-5.

[109] Loubère, R., Maire, P.-H., and Rebourcet, B. 'Chapter 13 - Staggered and Colocated Finite Volume Schemes for Lagrangian Hydrodynamics'. In: *Handbook of Numerical Methods for Hyperbolic Problems*. Ed. by Rémi Abgrall and Chi-Wang Shu. Vol. 17. Handbook of Numerical Analysis. Elsevier, 2016, pp. 319–352. DOI: https://doi.org/10.1016/bs.hna.2016.07.003.

[110] Lucy, L. B. 'A numerical approach to the testing of the fission hypothesis'. In: *The Astronomical Journal* 82.12 (1977), p. 1013. DOI: 10.1086/112164.

[111] Majdisova, Z. and Skala, V. 'A Radial Basis Function Approximation for Large Datasets'. In: *arXiv* June (2018).

[112] Mandel, J. 'Use of the Singular Value Decomposition in Regression Analysis'. In: *The American Statistician* 36.1 (1982), pp. 15–24.

[113] Manzari, M. T. 'Meshless Methods'. In: *Encyclopedia of Microfluidics and Nanofluidics*. Ed. by Dongqing Li. Boston, MA: Springer US, 2013, pp. 1–14.
DOI: 10.1007/978-3-642-27758-0_885-2.

[114] Mashwani, W. K., Haider, R.a, and Brahim Belhaouari, S. 'A Multi-swarm Intelligence Algorithm for Expensive Bound Constrained Optimization Problems'. In: *Complexity* 2021 (2021). Ed. by Atif Khan, pp. 1–18. DOI: 10.1155/2021/5521951.

[115] McDonald, D. B. et al. 'Global and local optimization using radial basis function response surface models'. In: *Applied Mathematical Modelling* 31.10 (2007), pp. 2095–2110.
DOI: https://doi.org/10.1016/j.apm.2006.08.008.

[116] Mckay, M., Beckman, R., and Conover, W. 'A Comparison of Three Methods for Selecting Vales of Input Variables in the Analysis of Output From a Computer Code'. In: *Technometrics* 21 (1979), pp. 239–245. DOI: 10.1080/00401706.1979.10489755.

[117] McNamara, G. R. and Zanetti, G. 'Use of the Boltzmann Equation to Simulate Lattice-Gas Automata'. In: *Physical Review Letters* 61 (20 1988), pp. 2332–2335. DOI: 10.1103/PhysRevLett.61.2332.

[118] Miettinen, K. *Nonlinear Multiobjective Optimization*. 1st Edition. Vol. 12. International Series in Operations Research & Management Science. Boston, MA: Springer Springer Science+Business Media, LLC, 1998, pp. 216–244. DOI: 10.1007/978-1-4615-5563-6.

[119] Minier, J.-P. and Pozorski, J. 'Derivation of a PDF model for turbulent flows based on principles from statistical physics'. In: *Physics of Fluids* 9.6 (1997), pp. 1748–1753. DOI: 10.1063/1.869291.

[120] Mirjalili, S., Jain, S. S., and Dodd, M. S. 'Interface-capturing methods for two-phase flows : An overview and recent developments'. In: *Center for Turbulence Research: Annual Research Briefs* 1 (2017), pp. 117–135. DOI: http://dx.doi.org/10.1080/10304312.2017.1409340.

[121] Molin, B. 'On the piston and sloshing modes in moonpools'. In: *Journal of Fluid Mechanics* 430 (2001), S0022112000002871. DOI: 10.1017/S0022112000002871.

[122] Molteni, D. and Colagrossi, A. 'A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH'. In: *Computer Physics Communications* 180.6 (2009), pp. 861–872. DOI: 10.1016/j.cpc.2008.12.004.

[123] Monaghan, J. J. 'Simulating free surface flows with SPH'. In: *Journal of Computational Physics* 110.2 (1994), pp. 399–406. DOI: 10.1006/jcph.1994.1034.

[124] Monaghan, J. J. 'Smoothed Particle Hydrodynamics'. In: *Annual Review of Astronomy and Astrophysics* 30.1 (1992), pp. 543–574. DOI: 10.1146/annurev.aa.30.090192.002551.

[125] Monaghan, J. J. 'Smoothed particle hydrodynamics'. In: *Reports on Progress in Physics* 68.8 (2005), pp. 1703–1759. DOI: 10.1088/0034-4885/68/8/R01.

[126] Monaghan, J. J. 'SPH compressible turbulence'. In: *Monthly Notices of the Royal Astronomical Society* 335.3 (2002), pp. 843–852. DOI: 10.1046/j.1365-8711.2002.05678.x.

[127] Monaghan, J. J. 'SPH without a Tensile Instability'. In: *Journal of Computational Physics* 159.2 (2000), pp. 290–311. DOI: 10.1006/jcph.2000.6439.

[128] Monaghan, J. J. and Rafiee, A. 'A simple SPH algorithm for multi-fluid flow with high density ratios'. In: *International Journal for Numerical Methods in Fluids* 71.5 (2013), pp. 537–561. DOI: 10.1002/fld.3671.

[129] Monaghan, J.J. and Kocharyan, A. 'SPH simulation of multi-phase flow'. In: *Computer Physics Communications* 87.1 (1995). Particle Simulation Methods, pp. 225–235. DOI: https://doi.org/10.1016/0010-4655(94)00174-Z.

[130] Montgomery, D. C. *Design and Analysis of Experiments*. 8th Edition. John Wiley & Sons, Incorporated, 2012, p. 730.

[131] Morris, J. P. 'Simulating surface tension with smoothed particle hydrodynamics'. In: *International Journal for Numerical Methods in Fluids* 33.3 (2000), pp. 333–353. DOI: 10.1002/1097-0363(20000615)33:3<333::AID-FLD11>3.0.CO;2-7.

[132] Morris, J. P., Fox, P. J., and Zhu, Y. 'Modeling low Reynolds number incompressible flows with curved boundaries using SPH'. In: *Journal of Computational Physics* 68.9 (1997), pp. 1173–1188. DOI: 10.1002/fld.2600.

[133] Nagasawa, M. 'Three Dimensional Hydrodynamical Simulation of Type II Supernova'. In: *Highlights of Astronomy: As presented at the XXth General Assembly of the IAU, 1988*. Ed. by Derek McNally. Dordrecht: Springer Netherlands, 1989, pp. 213–214. DOI: 10.1007/978-94-009-0977-9_33.

[134] Nayak, S. *Fundamentals of Optimization Techniques with Algorithms*. Elsevier, 2020. DOI: 10.1016/C2019-1-02539-9.

[135] Nelder, J. A. and Mead, R. 'A Simplex Method for Function Minimization'. In: *The Computer Journal* 7 (1965), pp. 308–313.

[136] Noh, W. F. and Woodward, P. 'SLIC (Simple Line Interface Calculation)'. In: *Lecture Notes in Physics (LNP)*. 1976, pp. 330–340. DOI: 10.1007/3-540-08004-X_336.

[137] Nvidia. *CUDA C++ programming guide*. 2024. URL: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html (visited on 2024).

[138] Nvidia. *CUDA zone*. 2024. URL: https://developer.nvidia.com/cuda-zone (visited on 2024).

[139] Offshore. *Modern offshore fleet comprised of same rig types as in the 1950s*. 2007. URL: https://www.offshore-mag.com/drilling-completion/article/16760817/modern-offshore-fleet-comprised-of-same-rig-types-as-in-the-1950s (visited on 2024).

[140] Olejnik, M. 'Modelling of Interfacial Flows with the Smoothed Particle Hydrodynamics Method'. PhD thesis. Institute of Fluid-Flow Machinery, Polish Academy of Sciences, 2019.

[141] Olejnik, M. and Pozorski, J. 'A Robust Method for Wetting Phenomena Within Smoothed Particle Hydrodynamics'. In: *Flow, Turbulence and Combustion* 104.1 (2020), pp. 115–137. DOI: 10.1007/s10494-019-00048-6.

[142] Olejnik, M., Szewc, K., and Pozorski, J. 'Modelling of the flow regime transition with the Smoothed Particle Hydrodynamics'. In: *ICMF-2016 – 9th International Conference on Multiphase Flow*. 3. 2016.

[143] Optimization, Gurobi. *Mathematical Optimization: Past, Present, and Future*. 2019. URL: https://www.gurobi.com/resources/mathematical-optimization-past-present-and-future-part-1/ (visited on 2024).

[144] Osher, S. and Sethian, J. A. 'Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations'. In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49. DOI: https://doi.org/10.1016/0021-9991(88)90002-2.

[145] Park, J. J. et al. 'Development of Design Technologies for Optimum Moonpool Shapes of Drill Ship'. In: *26th International Conference on Offshore Mechanics and Arctic Engineering*. ASMEDC, 2007, pp. 421–426. DOI: 10.1115/OMAE2007-29182.

[146] Parrilo, P. A. *Mathematical optimization techniques and their applications*. 2010. URL: https://www.icts.res.in/sites/default/files/1283168474PabloParrilo.pdf (visited on 2024).

[147] Patiño-Nariño, E. A. et al. 'Numerical study of single bubble rising dynamics for the variability of moderate Reynolds and sidewalls influence: A bi-phase SPH approach'. In: *Engineering Analysis with Boundary Elements* 129 (2021), pp. 1–26. DOI: 10.1016/j.enganabound.2021.04.011.

[148] Phadke, M.S. *Quality Engineering Using Robust Design*. Prentice-Hall International editions. Prentice Hall, 1989.

[149] Phillips, G. J. and Monaghan, J. J. 'A numerical method for three-dimensional simulations of collapsing, isothermal, magnetic gas clouds'. In: *Monthly Notices of the Royal Astronomical Society* 216.4 (1985), pp. 883–895. DOI: 10.1093/mnras/216.4.883.

[150] Poles, S. 'Meta-modeling with modeFRONTIER: Advantages and Perspectives'. In: *Newsletter EnginSoft modeFRONTIER Special Issue* (2008).

[151] Pozorski, J. and Olejnik, M. 'Smoothed particle hydrodynamics modelling of multiphase flows: an overview'. In: *Acta Mechanica* (2023). DOI: 10.1007/s00707-023-03763-4.

[152] Pozorski, J. and Wawrenczuk, A. 'SPH computation of incompressible viscous flows'. In: *Journal of Theoretical and Applied Mechanics* 40 (2002), pp. 917–937.

[153] Price, D. J. 'Smoothed particle hydrodynamics and magnetohydrodynamics'. In: *Journal of Computational Physics* 231.3 (2012), pp. 759–794. DOI: 10.1016/j.jcp.2010.12.011.

[154] Price, D. J. 'Smoothed Particle Hydrodynamics: Things I wish my mother taught me'. In: Cefalu, Sicily, Italy, 2011. DOI: 10.48550/arXiv.1111.1259.

[155] Prosperetti, A. and Tryggvason, G. *Computational Methods for Multiphase Flow*. Cambridge, UK: Cambridge University Press, 2007. DOI: 10.1017/CBO9780511607486.

[156] Rakhsha, M., Kees, C. E., and Negrut, D. 'Lagrangian vs. Eulerian: An analysis of two solution methods for free-surface flows and fluid solid interaction problems'. In: *Fluids* 6.12 (2021), pp. 1–27. DOI: 10.3390/fluids6120460.

[157] Ramakrishnan, S. 'Effect of the Moonpool on the Total Resistance of a Drillship'. MA thesis. A GPU program to compute SNP-SNP interactions in genome-wide association studies, 2013.

[158] Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. DOI: 10.7551/mitpress/3206.001.0001.

[159] Raza, W., Ma, S.-B., and Kim, K.-Y. 'Multi-Objective Optimizations of a Serpentine Micromixer with Crossing Channels at Low and High Reynolds Numbers'. In: *Micromachines* 9.3 (2018), p. 110. DOI: 10.3390/mi9030110.

[160] Read, J. I., Hayfield, T., and Agertz, O. 'Resolving mixing in smoothed particle hydrodynamics'. In: *Monthly Notices of the Royal Astronomical Society* 405.3 (2010), pp. 1513–1530. DOI: 10.1111/j.1365-2966.2010.16577.x.

[161] Renardy, M. et al. 'To Sobol or not to Sobol? The effects of sampling schemes in systems biology applications'. In: *Mathematical Biosciences* 337 (2021), p. 108593. DOI: 10.1016/j.mbs.2021.108593.

[162] Rezavand, M., Taeibi-Rahni, M., and Rauch, W. 'An ISPH scheme for numerical simulation of multiphase flows with complex interfaces and high density ratios'. In: *Computers and Mathematics with Applications* 75.8 (2018), pp. 2658–2677. DOI: 10.1016/j.camwa.2017.12.034.

[163] Rezavand, M., Zhang, C., and Hu, X. Y. 'A weakly compressible SPH method for violent multi-phase flows with high density ratio'. In: *Journal of Computational Physics* 402 (2020), p. 109092. DOI: 10.1016/j.jcp.2019.109092.

[164] Rockafellar, T. R. *Fundamentals of Optimization*. Tech. rep. Seattle: University of Washington, 2007, p. 133.

[165] Rosato, Dominick and Rosato, Donald. '5 - Computer-aided design'. In: *Plastics Engineered Product Design*. Ed. by D. Rosato and D. Rosato. Amsterdam: Elsevier Science, 2003, pp. 344–380. DOI: https://doi.org/10.1016/B978-185617416-9/50006-5.

[166] Russo, R. et al. *An Aerodynamic Design Framework based on Algorithmic Differentiation*. Vol. 102. March. 2015, pp. 10–16.

[167] Ryskin, G. and Leal, L. G. 'Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid'. In: *Journal of Fluid Mechanics* 148 (1984), pp. 19–35. DOI: 10.1017/S0022112084002226.

[168] Ryskin, G. and Leal, L. G. 'Orthogonal mapping'. In: *Journal of Computational Physics* 50.1 (1983), pp. 71–100. DOI: https://doi.org/10.1016/0021-9991(83)90042-6.

[169] Sacks, J. et al. 'Design and Analysis of Computer Experiments'. In: *Statistical Science* 4.4 (1989), pp. 409–423.

[170] Schoenberg, I. J. 'Contributions to the problem of approximation of equidistant data by analytic functions'. In: *Quarterly of Applied Mathematics* 4.1 (1946), pp. 45–99.

[171] Shah, K. Y., Cumberbatch, H., and Etchart, L. 'Optimization Techniques using modeFRONTIER - Process Review'. In: *Proceedings of the 34th Florida Conference on Recent Advances in Robotics*. Tampa, FL: University of South Florida, 2021. DOI: 10.5038/GVXG7608.

[172] Silveira, C. L. B. et al. 'Mathematical optimization versus Metaheuristic techniques: A performance comparison for reconfiguration of distribution systems'. In: *Electric Power Systems Research* 196.March (2021), p. 107272. DOI: 10.1016/j.epsr.2021.107272.

[173] Sobol', I. M. 'On the distribution of points in a cube and the approximate evaluation of integrals'. In: *USSR Computational Mathematics and Mathematical Physics* 7.4 (1967), pp. 86–112. DOI: https://doi.org/10.1016/0041-5553(67)90144-9.

[174] Soligo, G., Roccon, A., and Soldati, A. 'Turbulent Flows With Drops and Bubbles: What Numerical Simulations Can Tell Us—Freeman Scholar Lecture'. In: *Journal of Fluids Engineering* 143.8 (2021). DOI: 10.1115/1.4050532.

[175] Son, H.-J. et al. 'Drag Reduction of Recess Type Moonpool Under Vessel's Forward Speed'. In: *Volume 1: Offshore Technology*. Vol. 1. ASMEDC, 2008, pp. 143–148. DOI: 10.1115/OMAE2008-57118.

[176] SpA, ESTECO. *ESTECO*. 2024. URL: https://www.esteco.com (visited on 2024).

[177] Spricigo, E. and Pozorski, J. 'Simulation of sloshing in the moon pool with Smoothed Particle Hydrodynamics'. In: *Journal of Physics: Conference Series* 2367.1 (2022), p. 012030. DOI: 10.1088/1742-6596/2367/1/012030.

[178] Stellingwerf, R. F. and Peterkin, R. E. 'Smooth particle magnetohydrodynamics'. In: *Technical report, Albuquerque: Mission Res. Corp.* (1990).

[179] Sun, P. et al. 'Investigation of Coalescing and Bouncing of Rising Bubbles Under the Wake Influences Using SPH Method'. In: *Volume 8B: Ocean Engineering*. Vol. 8B. American Society of Mechanical Engineers, 2014. DOI: 10.1115/OMAE2014-23789.

[180] Sussman, M., Smereka, P., and Osher, S. 'A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow'. In: *Journal of Computational Physics* 114.1 (1994), pp. 146–159. DOI: https://doi.org/10.1006/jcph.1994.1155.

[181] Swegle, J. W., Hicks, D. L., and Attaway, S. W. 'Smoothed Particle Hydrodynamics Stability Analysis'. In: *Journal of Computational Physics* 116.1 (1995), pp. 123–134. DOI: https://doi.org/10.1006/jcph.1995.1010.

[182] Szewc, K. 'Hydrodynamics approach for modelling of multiphase flows with interfaces'. PhD thesis. Institute of Fluid-Flow Machinery, Polish Academy of Sciences, 2013.

[183] Szewc, K. and Lewandowski, M. T. 'Further investigation of the spurious interface fragmentation in multiphase Smoothed Particle Hydrodynamics'. In: *arXiv* 4 (2016), pp. 1–10.

[184] Szewc, K., Pozorski, J., and Minier, J.-P. 'Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method'. In: *International Journal for Numerical Methods in Engineering* 92.4 (2012), pp. 343–369. DOI: 10.1002/nme.4339.

[185] Szewc, K., Pozorski, J., and Minier, J.-P. 'Simulations of single bubbles rising through viscous liquids using Smoothed Particle Hydrodynamics'. In: *International Journal of Multiphase Flow* 50 (2013), pp. 98–105. DOI: 10.1016/j.ijmultiphaseflow.2012.11.004.

[186] Szewc, K., Pozorski, J., and Minier, J.-P. 'Spurious interface fragmentation in multiphase SPH'. In: *International Journal for Numerical Methods in Engineering* 103.9 (2015), pp. 625–649. DOI: 10.1002/nme.4904.

[187] Tafuni, A. et al. 'A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models'. In: *Computer Methods in Applied Mechanics and Engineering* 342.August (2018), pp. 604–624. DOI: 10.1016/j.cma.2018.08.004.

[188] Takewaki, H., Nishiguchi, A., and Yabe, T. 'Cubic interpolated pseudo-particle method (CIP) for solving hyperbolic-type equations'. In: *Journal of Computational Physics* 61.2 (1985), pp. 261–268. DOI: https://doi.org/10.1016/0021-9991(85)90085-3.

[189] Tryggvason, G., Scardovelli, R., and Zaleski, S. *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*. Cambridge University Press, 2011, pp. 313–338. DOI: 10.1017/CBO9780511975264.

[190] Unverdi, S. O. and Tryggvason, G. 'A front-tracking method for viscous, incompressible, multi-fluid flows'. In: *Journal of Computational Physics* 100.1 (1992), pp. 25–37. DOI: https://doi.org/10.1016/0021-9991(92)90307-K.

[191] Verbrugghe, T. et al. 'Wave generation and absorption via SPH inlet / outlet conditions'. In: *SPHERIC International Workshop 2019*. June. 2019.

[192] Violeau, D. and EDF R&D. *Fluid Mechanics and the SPH Method: Theory and Applications*. 1st ed. Oxford: Oxford, University Press, 2012, pp. 1–616.
DOI: 10.1093/acprof:oso/9780199655526.001.0001.

[193] Violeau, D. and Fonty, T. 'Calculating the smoothing error in SPH'. In: *Computers & Fluids* 191 (2019), p. 104240. DOI: https://doi.org/10.1016/j.compfluid.2019.104240.

[194] Wendland, H. 'Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree'. In: *Advances in Computational Mathematics* 4.1 (1995), pp. 389–396.
DOI: 10.1007/BF02123482.

[195] Xenakis, A. M. et al. 'An incompressible SPH scheme with improved pressure predictions for free-surface generalised Newtonian flows'. In: *Journal of Non-Newtonian Fluid Mechanics* 218 (2015), pp. 1–15. DOI: 10.1016/j.jnnfm.2015.01.006.

[196] Xu, R., Stansby, P., and Laurence, D. 'Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach'. In: *Journal of Computational Physics* 228.18 (2009), pp. 6703–6725.
DOI: https://doi.org/10.1016/j.jcp.2009.05.032.

[197] Y.M. Lo, E. and Shao, S. 'Simulation of near-shore solitary wave mechanics by an incompressible SPH method'. In: *Applied Ocean Research* 24.5 (2002), pp. 275–286. DOI: https://doi.org/10.1016/S0141-1187(03)00002-6.

[198] Yabe, T., Xiao, F., and Utsumi, T. 'The Constrained Interpolation Profile Method for Multiphase Analysis'. In: *Journal of Computational Physics* 169.2 (2001), pp. 556–593.
DOI: https://doi.org/10.1006/jcph.2000.6625.

[199] Yan, J. et al. 'Updated Lagrangian Particle Hydrodynamics (ULPH) modeling and simulation of multiphase flows'. In: *Journal of Computational Physics* 393 (2019), pp. 406–437.
DOI: 10.1016/j.jcp.2019.05.017.

[200] Youngs, D. 'Time-Dependent Multi-material Flow with Large Fluid Distortion'. In: *Numerical Methods in Fluid Dynamics* 24 (1982), pp. 273–285.

[201] Zainali, A. et al. 'Numerical investigation of Newtonian and non-Newtonian multiphase flows using ISPH method'. In: *Computer Methods in Applied Mechanics and Engineering* 254 (2013), pp. 99–113. DOI: 10.1016/j.cma.2012.10.005.

[202] Zhang, A-M., Sun, P., and Ming, F. 'An SPH modeling of bubble rising and coalescing in three dimensions'. In: *Computer Methods in Applied Mechanics and Engineering* 294 (2015), pp. 189–209. DOI: 10.1016/j.cma.2015.05.014.

[203] Zheng, B. X. et al. 'Multiphase smoothed particle hydrodynamics modeling of forced liquid sloshing'. In: *International Journal for Numerical Methods in Fluids* 93.2 (2021), pp. 411–428. DOI: 10.1002/fld.4889.

[204] Zou, S. and Dalrymple, R. A. 'Sediment suspension simulation under oscillatory flow by SPH-SPS method'. In: *Coastal Engineering 2006*, pp. 2477–2484. DOI: 10.1142/9789812709554_0209.

CPU and GPU A

In this chapter, we aim to recall some general information about the CPU and GPU, that are useful to know when interfacing our in-house code.

A.1 Introduction to CPU and GPU

The CPU[1], Central Processing Unit, is the part of a computer that executes instructions, as explained by Rosato and Rosato [165], and it is usually referred to simply as the processor. CPUs are well-suited for complex decision-making, such as running operating systems, handling user interfaces, and managing overall system functionality. The GPU[2], Graphics Processing Unit, is a device consisting of many cores (thousands) designed to process multiple tasks concurrently. The GPUs were firstly designed to accelerate the processing of images and videos; nowadays they are also utilises to handle a wide range of parallel tasks, including processing large amounts of data and running scientific simulations.

1: More information about the CPU in the web page CPU [48].

2: More information about the GPU in the web page GPU [49].

CPUs and GPUs complement each other in a computer system: the first ones manages the overall activities, and the second ones are usually implemented to solve problems that involve large amounts of data simultaneously (such as training of neural networks).

Three leading companies specialised in GPUs for parallel computing are: Intel, AMD (Advanced Micro Devices) and Nvidia. Their products use a specific architecture and programming language to work with the parallel process. We mostly focus on Nvidia GPUs features because our code has been written specifically to work with with Nvidia graphic cards. One of the most important tool provided by Nvidia is CUDA[3]. The acronym CUDA stands for Compute Unified Device Architecture, and it is a parallel computing platform and application programming interface (API) model created by Nvidia[4]. CUDA's library comprises the compiler named *nvcc* and the so-called kernel functions, compatible also with C and C++ programming languages. The kernel functions are key concepts in CUDA, they are executed in parallel and are used to organize the parallelisation hierarchy.

3: Definition of CUDA e some applications in CUDA zone [138].

4: An alternative to CUDA is OpenCL, this framework is not provided by Nvidia and it is meant for parallel computing. Moreover, it allows the developers to write programs for different types of processors, including CPUs, GPUs, and other accelerators.

In Fig. A.1, CPU and GPU architectures are compared. While CPUs typically have a few powerful cores optimized for sequential processing, GPUs consist of many smaller, specialized cores. In the GPU, more transistors are devoted to data processing rather than data caching and flow control, which is beneficial for parallel computations. Specifically, in Fig. A.1 on the right, a basic building block composed of local cache memory (purple), control logic unit (yellow) and a set of cores (green) is called streaming multiprocessor (SM). Each SM contains multiple CUDA

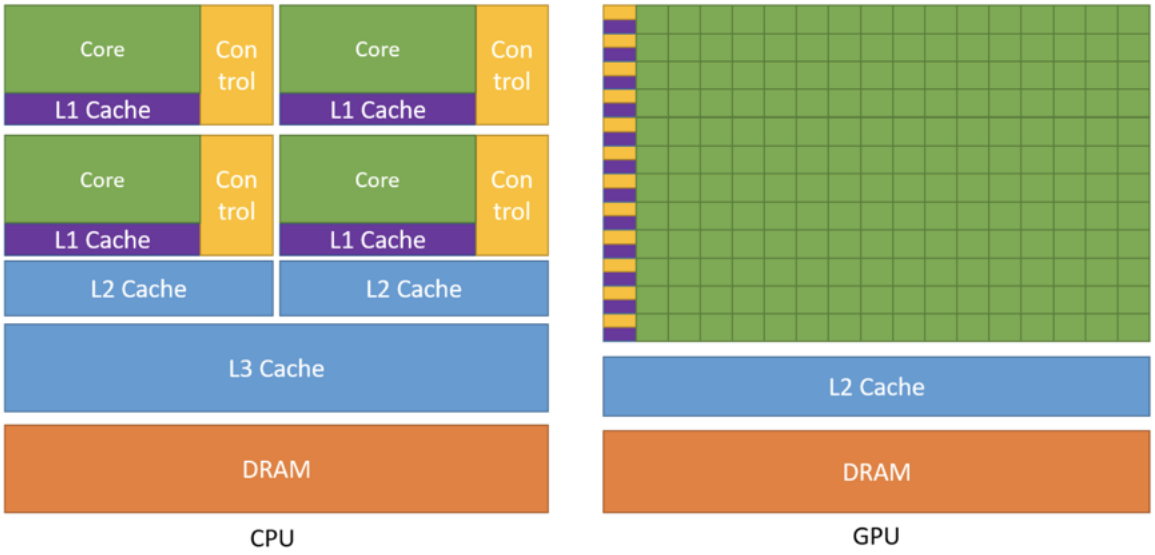


Figure A.1: CPU and GPU architectures compared, taken from Nvidia [137].

cores and is where the actual computation takes place. The number of SMs in a GPU is a crucial factor in determining its parallel processing capability. CUDA cores are fundamental processing units within an SM, each of them acts independently. Modern GPUs can have thousands of CUDA cores, in fact they determine the parallel processing power.

A.2 GPU architecture

Based on how the labour is partitioned between the CPU and GPU, in CUDA programming, it has been decided to name the CPU "host" and the GPU "device". The host-device system follows this process: the host sends the data to the device and calls the kernel functions which activate the parallel execution on the GPU; the computation runs on the device and finally gives results that are sent back to the host.

In Fig. A.2 we see a representation of the CUDA architecture [8]. When the host calls the kernel function, this function is performed a set of instructions on each data element, with each data element occupying a CUDA computing unit named thread. As explain in the thesis by Ramakrishnan [157], a consistent number of threads is collected in set to form a thread block, up to 1024 on the latest hardware, a set of blocks composes a grid (see also [74]). The blocks are arranged in into a one-dimensional, two-dimensional, or three-dimensional grids; the blocks number per grid is related to the computational effort required to process the data, it may be not necessarily to use all of the thread. This structure is rather efficient because the kernel functions can be simultaneously processed by multiple threads, this number can amount up to the value obtain multiplying the threads per block times the number

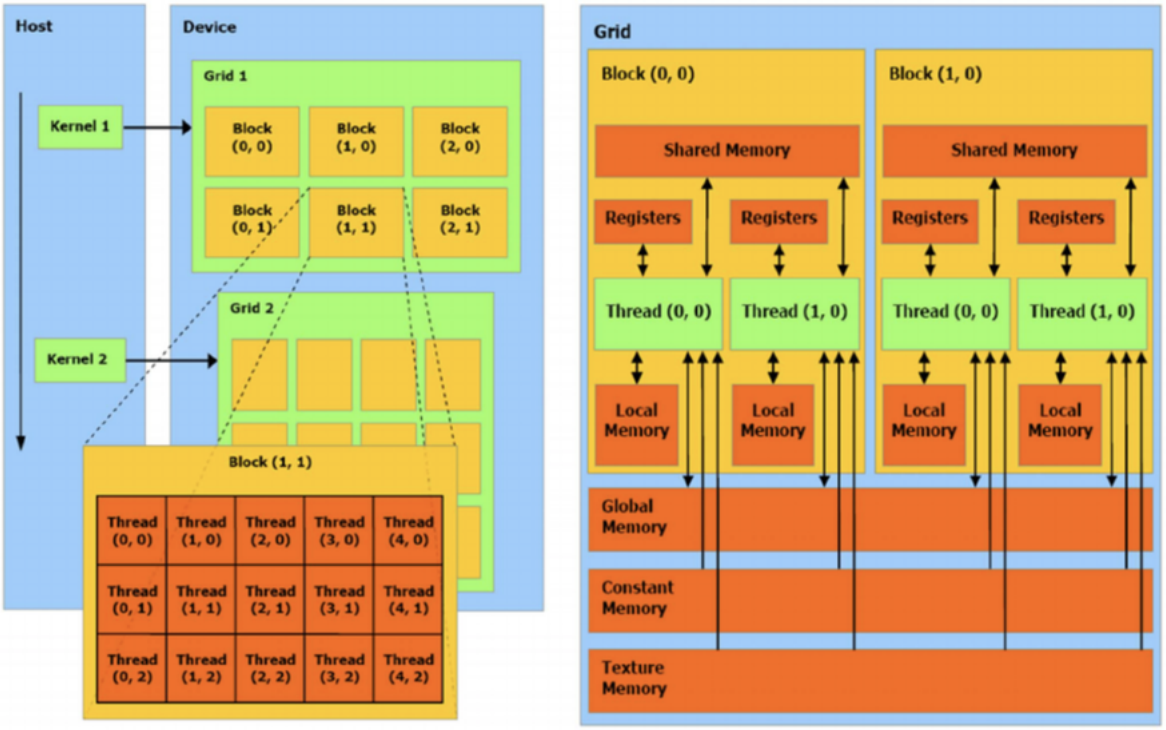


Figure A.2: Left: CUDA architecture; right: CUDA memory model. Image from CUDA architecture [8].

of blocks. Obviously, it is important to exploit the block in a way that the largest amount of threads are computed per loop. This optimisation is established by GPU scheduler, which organizes threads into warps that are clusters of 32 threads each; the threads that belong to a certain warp are force to execute all the same instruction. At a higher level, each SM processes multiple warps in a round-robin fashion. If threads within a warp follow different execution paths (e.g., due to conditional statements), the GPU executes both paths but uses predication to mask out threads that are not following the same path. This avoids performance penalties associated with thread divergence.

The management and optimization of memory access patterns are crucial for achieving high performance. Developers need to consider factors such as data locality, memory coalescing, and minimizing global memory accesses to ensure efficient usage of the memory and reduce possible latency. Data movement between different types of memory, such as transferring data from the host (CPU) to the device (GPU) and vice versa, is also an important aspect of CUDA programming, and it is done explicitly by the programmer. Efficient use of memory plays a significant role in maximizing the parallel processing efficiency of the GPU.

In CUDA there are six main types of memory: unified, texture, constant, global, shared and local together with register. The hierarchy that leads the memory usage is the following: any thread has its own register which is a small, fast, on-chip memory location; whenever the register is exhausted, the thread has also access to its local memory. Registers are

used for storing local variables and intermediate values within a thread [74]. All the threads of a single block refer to the shared memory, which is fast and low-latency that exists for the duration of a block's execution. The data that have to be accessible by all threads of the device are stored in the global memory, which persists throughout the execution of the CUDA program and is the largest memory available for threads. The global memory is read-write memory, the constant and texting memory are also accessible by all the threads but they are slow and read-only memory. The first one is used to store the constant data, the second is primarily used in graphics processing. Finally, CPU and GPU share the unified memory, which persists throughout the execution of the CUDA program.

A.3 CUDA programming language key features

The programming in CUDA does not consist in a new language, but rather on a series of commands and functions that extend the existing library of a language, such as C or C++. Any program that implement CUDA follows three main steps[5]:

5: Some information of how to properly program these three steps in CUDA Refresher [69].

1. the input data are copied from host memory and pasted to device memory, this transfer is referred as host-to-device;
2. the GPU program is loaded and executed, data are cached on chip;
3. finally the so-called device-to-host transfer, in which results are transferred from device memory to host memory.

Every CUDA kernel is initiated by the `__global__` declaration specifier; more keywords are threadIdx, blockIdx and blockDim which give access to thread and block indices and blocks dimension respectively. The declarations `__device__` and `__constant__` define in which type of memory the data should be stored.

The code: cuSPH2d B

In this chapter, we aim to present an overview of the code: the circumstance under it was designed and created, its main characteristics, the utilization of GPU computing, and how it has been developed through the years.

B.1 The code

At this point of the manuscript, it is already well-known why SPH is an possible choice for multiphase flows: its inherent structure, made of particles, easily defines the position of the interface. The first version of the code was designed to handle some of the most important benchmarks for multiphase SPH, e.g. lid-driven cavity, Taylor-Green vortex, Rayleigh-Taylor instability, dam-break etc., but even for more complex test cases as dispersed flow, rising bubble, applications with surfactants and more.

The first and main author is Dr. K. Szewc, who started this project back in 2014. Since then, the code has been developed by other Ph.D. students, including Dr. M. Olejnik, who implemented the code for wetting phenomena [141] and two-phase flows in channels [142].

The code has been written in `C++` language and parallelized with NVIDIA® CUDA® . All the SPH simulations presented in these paper ran on NVIDIA GPU GeForce® RTX 2080 Ti. In these conditions, for the case of moon pool flow, presented in Chapter 5, an SPH simulation has revealed to be about five times faster than an Ansys® Fluent® simulation with comparable resolution, computed in a Windows 10 Pro with Intel® Core™ i5-3570 CPu @ 3.40 GHz, 12 GB RAM.

B.2 Why implement the SPH code in the GPU

The SPH approach is renowned for its demand for resource-intensive code, it is crucial to enhance the computational speed to increase the usability of the code. We can outline few valid reasons that can lead the reader to lean towards a GPU implementations for SPH algorithms.

A generic SPH domain is discretised by a certain, and sometimes substantial, number of particles that eventually move and vary position at each time step. The computing ability of the GPU allows to treat simultaneously the particles, accelerating the calculation related to each particle performing in parallel rather than in serial. Additionally, GPUs massive parallel power allows for relatively large scale problem, which is particularly useful considering that SPH resolution is directly correlated

with the number of particles. Moreover, the high memory bandwidth of GPUs can contribute to improve the performance, allowing for faster data transfer between the CPU and GPU. In Chapter 2, we have emphasized the benefits of utilizing the explicit time integration scheme in the SPH algorithm. The explicit time integration scheme is an additional incentive for opting for the GPU parallelization: this calculation is done independently for each particle at any time step. Moreover, as pointed out in [74], the interaction between particles is pairwise: for weakly compressible SPH the forces calculation does not require matrix operations, but rather the estimation of relative position and velocity between a pair of particles. The GPU implementation can only benefits from this approach.

In the work by Crespo et al. [41], developed and improved later in Domínguez et al. [51, 50], the single particle pair-wise interactions are demonstrated to be notably advantageous when executed by a single thread. Specifically, the search for particle neighbour and the forces computations is extremely time demanding if done entirely on the CPU, since this must be done for each single particle in the domain. Likewise, in [41, 51, 50] it is proved that implementing the main loop partially in the GPU (and completing it in the CPU) is not convenient too because the transfer of data slows down the overall simulation. The best choice is to entirely implement the particle interaction procedure and the forces calculation on the GPU, using the shared memory, which is significantly faster compared to global memory.

B.3 Particle search algorithm

There exist two ways to improve the efficiency of an SPH code: implement smoothing kernel with compact support and optimise the search of the neighbour particles.

If the first condition can be satisfied by opting for most convenient kernel as explained in Chapter 2, the second condition opens the door to a new discussion about particles interaction. Linking a particle to every other single particle in the domain is time-consuming: the use of kernel compact support eases the process, hence only particles falling within the influence of the kernel contribute to the interpolation. The search for neighbours, particles that can be included in the kernel interpolation, is dull when implemented to the entire domain. Preferring a local search can save a lot of time.

In our code, the local search for neighbours is facilitated by the auxiliary grid as in Fig. B.1: the entire computational domain is divided in square-shape cells of equal size. For ease, the grid is created at every time step and its cell's size matches the range of the kernel (specifically, for the Wendland kernel); each particle can only interact with a restricted number of particles situated in the central cell and in the adjacent eight

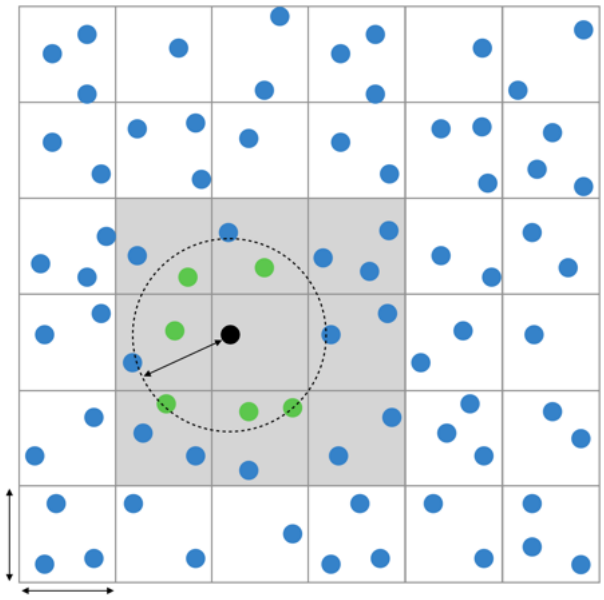


Figure B.1: Representation of the auxiliary grid, useful to optimise the search of neighbour particles in the computational domain. The particle-interaction method consists of three main steps: (1) Any particle is given an hash index based on it position on the grid. (2) The particles are sorted on the basis on the hash index. (3) When calculating the interactions for a certain particle, which coloured in black in the figure, the search of the neighbours is restricted to the particles included in the central cell (the one the black particle belongs to) and the surrounding eight cells, in figure coloured in grey. This approach eases the search for the green particles, that fall under the influence of the kernel.

cells (in two-dimensions). The algorithm for searching particles inside the grid cells has two steps: the first one is to give the particle a new hash index; then the particles are sorted based on their hash value. The hash function is applied to the particle's position, it converts the continuous space into discrete indices, mapping particles to specific cells in the grid: particles in the same, or in the eight nearby cells, are considered neighbours, they are engaged in interactions needed to calculate the values of forces. This approach speeds up the calculations, reducing the overall computational cost.

B.4 The code general scheme

The code can be represented by the simplified flowchart in Fig. B.2, in which are included the primary steps executed by the code.

The iterative process starts with the initial step of reading the commands on the terminal, and compiling the entire program. The host systematically retrieves the characteristics of each particle, including their positions, properties and parameters related to the domain. After this phase of data acquisition, the GPU is selected for computation; the dedicated memory on the GPU is allocated to accommodate the incoming data from the CPU. The initiation of data migration from the host to the device sets the basis for the computation. The user must also define the time interval for storing the data. When the calculation is completed, the results are transferred from the GPU to the CPU to be recorded to a designated text file. As the simulation concludes, the procedure

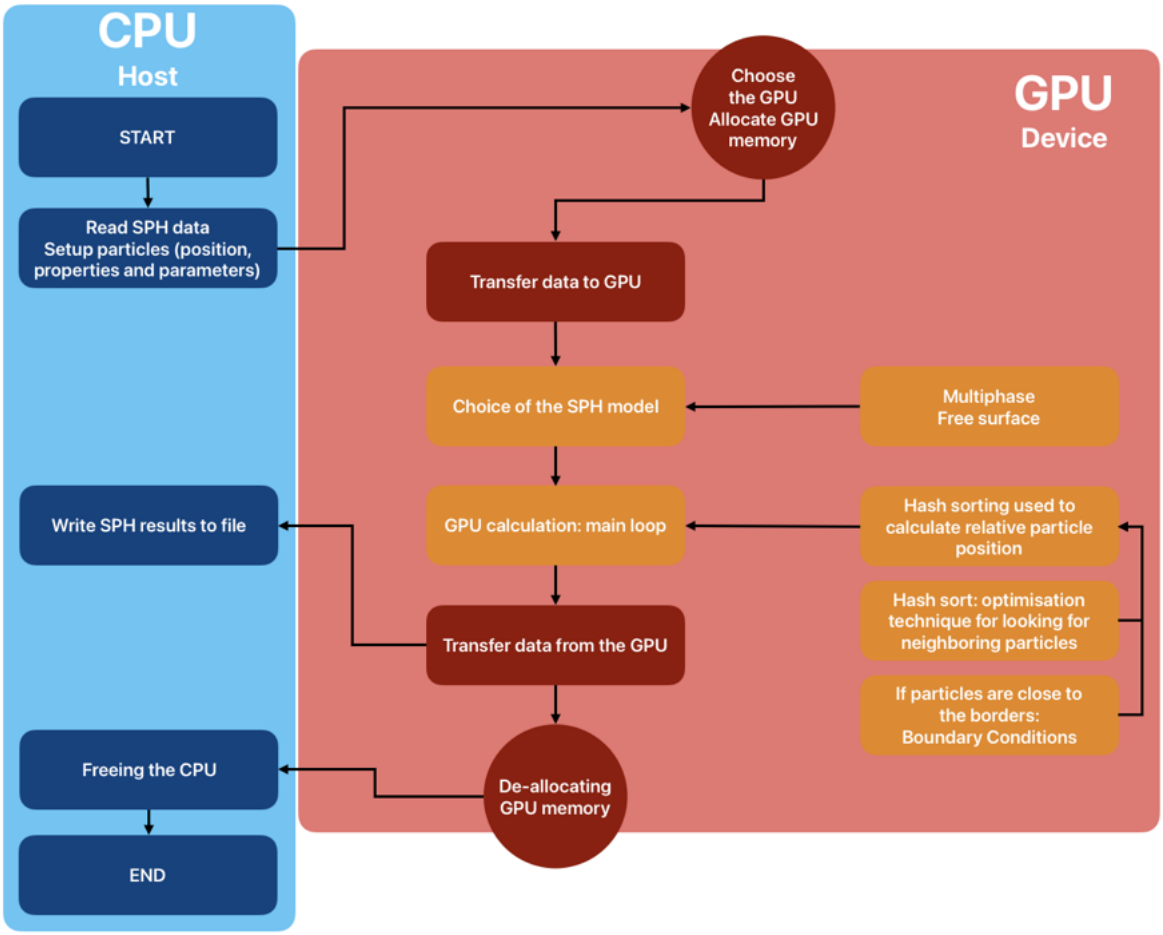


Figure B.2: Simplified flowchart of the code

ensures the de-allocation of the GPU, releasing its memory, as well as the freeing of CPU. The choice of the model depends on the type of flow, be it multiphase, free-surface, dispersed flow, etc. In our code, we have incorporated equations suggested by Hu and Adams [87], along with various mixed models, including those integrating the novel mVER formulation discussed in Chapter 3. Naturally, the main loop implements the hash sorting procedure, with particular attention when particles are near the boundary.

The boundary conditions in the code

C

In SPH, the boundaries are usually treated with two main techniques as described in Chapter 2: with fixed particles or with ghost particles (also called mirror particles). While the first method is suitable for cases involving complex geometries, the second is more advantageous as it computes fewer particles. Our code has been validated for the ghost particles method.

C.1 Ghost particles BC in the code

When particle a that approaches the borders of the domain in the position $\mathbf{r}_a$, that is to be considered as an straight line, the code creates an image of this particle a' located at

$$\mathbf{r}_{a'} = 2\mathbf{d} + \mathbf{r}_a \tag{C.1}$$

where $\mathbf{d}$ is the vector pointing perpendicularly from the particle to close wall, as illustrated in Fig. C.1.a. The layer composed by the particles images is finite, defined in a way to fulfil the kernel support and prevent the kernel truncation.

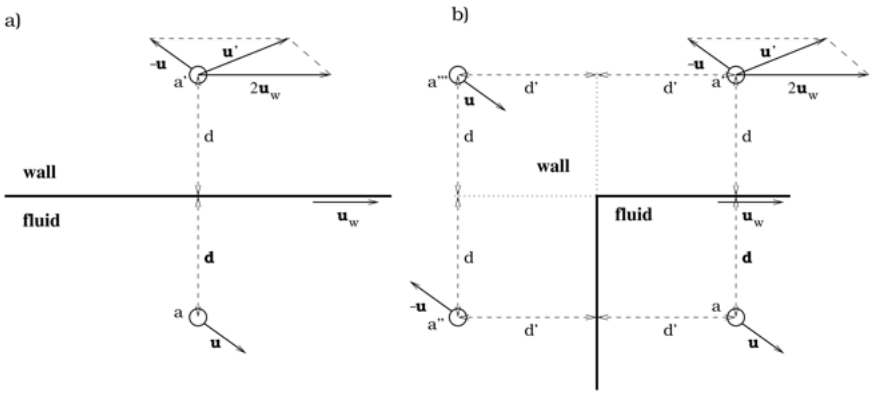


Figure C.1: Representation of the no-slip condition for ghost-particle from [184]: a) linear wall, b) inner corner.

The velocity formulation for mirrored particle in the case of no-slip wall condition is

$$\mathbf{v}_{a'} = 2\mathbf{v}_w - \mathbf{v}_a \tag{C.2}$$

where $\mathbf{v}_w$ is the velocity of the wall. To ensure the correct imposition of the Neumann boundary condition on pressure, we impose the following simplification

$$\frac{\partial P}{\partial n} = 0 \tag{C.3}$$

where $\mathbf{n}$ is the normal vector to the boundary. The particle properties

that allow for mirroring effect are

$$m_{a'} = m_a, \qquad \rho_{a'} = \rho_a \qquad P_{a'} = P_a. \tag{C.4}$$

When the particle is close to an inner corner of the domain, as in Fig. C.1.b, although three mirror particles are required to deal with the symmetry, they are not enough to guarantee the correctness of the continuity equation. To prevent any penetration of the boundary by the particles, we must add a repulsive force, which in WCSPH is done increasing the local density. Assuming a statistically homogeneous distribution of particle positions, the divergence of velocity is about zero in the proximity of the corner, therefore there is no observable change in density and, as a consequence, in pressure too. This issue is addressed by implementing a free-slip condition in the computation of the divergence of velocity. Further details are available in the paper by Szewc et al. [184] (2012). The numerical implementation of the ghost-particle boundary approach is notably straightforward, the interaction between fluid particles and mirror wall particles is self-regulating, and the ghost particles do not affect the overall calculation.