

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY
AND MEDICINE

DEPARTMENT OF EARTH SCIENCE AND ENGINEERING

PHD THESIS

Smoothed Particle Hydrodynamics for Fluid-Solid coupling: Modelling fixed and mobile boundaries

Author:

Anton ESMail-YAKAS
ae1212@ic.ac.uk

First Supervisor:

Prof. Stephen NEETHLING

Second Supervisor:

Prof. Gareth COLLINS

April 3, 2023

Abstract

Smoothed Particle Hydrodynamics (SPH) is a meshless Lagrangian numerical method which has undergone extensive development in recent years. SPH has properties which make it especially suited to certain problem types with which traditional methods have struggled. This includes fluid-solid coupled problems, which are particularly relevant for the modelling of coastal dynamics - a domain which is becoming increasingly relevant owing to the changing global infrastructure and climate. SPH, however, is a relatively young method, and suffers drawbacks which have been dubbed its grand challenges. The main one of interest for this work, is the handling of boundary conditions within the method. This thesis aims to present a discussion on SPH in the context of modelling boundaries, both fixed and mobile. Improvements were made to the so-called semi-analytical boundary method, solving the mathematical and numerical problems associated with the method, and presenting the work in such a way that it can easily be ported to existing SPH models. Further discussions and adjustments were also made to the boundary method, attempting to tackle some of its inconsistencies and render it more viable for typical use cases. Finally, the coupling of the DEM method for modelling mobile boundaries was addressed, with a particular focus on mesoscale modelling of solids at a similar resolution to the fluid domain. This was previously an unviable problem, but with improvements to the solid fraction calculation presented here, this is no longer the case. This presents the opportunity for full range mesoscale modelling in SPH.

Originality declaration

I hereby declare that the work in this thesis is all original and my own, except where duly acknowledged and appropriately referenced.

Anton Esmail-Yakas

June 2022

Copyright declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

Working in a relatively new domain, and particularly with the current global developments limiting departmental contact, it was often very difficult to find professional support. Nonetheless, my supervisor Professor Stephen Neethling has been invaluable in helping me get up to speed with our SPH code base and help get me to a level where I can confidently handle it myself. He is always willing to help and keenly curious about how we can improve the code, which is a great energy to be around. Dr Larry Kostorz, though not an expert in SPH, is an expert in mathematics and was extremely helpful through the central parts of my PhD. It is to him I owe credit that I have co-authored 3 published papers, and I imagine the path of this PhD would have been very different had it not been for him.

The completion of this PhD has been a particularly long and arduous process, not at all helped by the global situations occurring at the same time. As such, my non-professional help away from the university has been by far the most impactful for me. All of my family and my now wife did their absolute best to help get me through this. As difficult as it was at times, Baba, Helena, and of course Mama, have stuck by me and helped me reach my final goal.

Contents

1	Introduction	9
1.1	Coastal Hazards	9
1.2	Fluid-Solid Coupling	11
1.2.1	Coupling mechanisms in a coastal wave event	11
1.3	Numerical modelling for Fluid-Solid Coupling	16
1.3.1	Discrete Element Method	18
1.3.2	Smoothed Particle Hydrodynamics	18
1.4	Outline of Thesis	20
2	Smoothed Particle Hydrodynamics Method	22
2.1	Conceptual Framework	23
2.2	Mathematical Formulation	24
2.3	Kernel function and accuracy of the method	26
2.4	Derivatives in SPH	29
2.5	Fluid Dynamics equations	32
2.6	Further techniques	35
2.6.1	Density and Pressure	35
2.6.2	Viscosity	43
2.6.3	Particle Shifting	45
2.6.4	XSPH	47
2.7	Computational implementation	48
2.7.1	Time integration schemes	50
2.7.2	Parallel framework	53
2.7.3	Computational algorithm	54
3	Wall boundaries	57
3.1	Repulsive containment and leaked particles	58
3.2	Ghost particles	62
3.2.1	Reflected particles	67
3.2.2	Fixed particles	68
3.3	Semi-analytical boundary method	69
4	An improved semi-analytical boundary method (1)	75
4.1	Summary of the method	78
4.1.1	Two dimensions	78

4.1.2	Three dimensions	80
4.2	Simple planar Boundaries in 3D	83
4.2.1	Piecewise-linear segment	84
4.2.2	Circular arc segment	85
4.3	Spherical boundaries in 3D	86
4.3.1	Case (a): $\alpha = 0$	88
4.3.2	Case (b): $0 < \alpha \leq \beta - 1$	89
4.3.3	Case (c): $\beta - 1 < \alpha < \beta + 1$	90
4.3.4	Case $\beta < 1$ - Boundaries larger than SPH particles	91
4.4	Closed-form solution of integrals	92
4.5	Implementation in an SPH context	95
4.5.1	Evaluating $\gamma_j, \nabla \gamma_j, A_j$	96
5	An improved semi-analytical boundary method (2)	108
5.1	Governing equations	108
5.2	Further considerations	111
5.3	Preliminary results	117
6	Mobile boundaries	128
6.1	Fluid-Solid coupling	128
6.2	Large-body DEM	131
6.3	Small-body DEM	133
6.3.1	Fluid particle equations	133
6.3.2	Solid body equations	134
6.3.3	Inclusion of a lift-force for solid particle saltation mechanics	138
6.3.4	Semi-analytical calculations of porosity	138
7	Conclusion	146
7.1	Future work	149
8	References	153
A	Full expansion of $\frac{\partial^2 A}{\partial \mathbf{r}^2} : (\mathbf{r} - \mathbf{r}') \otimes (\mathbf{r} - \mathbf{r}')$	166
B	Derivation of dimension reduction process for γ in 2D	166

List of Figures

1	Smoothing of discrete values to a continuous field.	24
2	Cubic spline and Quintic Wendland function and their derivatives.	28
3	SPH interpolation near a free surface.	36
4	Comparison of pressure initialisation vs no initialisation for a hydrostatic box of fluid.	44
5	Neighbour cell stencil for SPH interaction loop.	51
6	Truncated kernel support near a wall boundary.	58
7	Kernel support near a wall boundary containing wall particles.	63
8	Diagram of fixed and reflected wall particle boundary methods [161].	66
9	Velocity assignment of reflected boundary particles for no-slip and free-slip conditions [13].	68
10	Integration domain of an SPH particle support intersecting a wall boundary.	76
11	Spherical isosurface of a radial function in the vicinity of a straight boundary edge.	85
12	Intersection of an SPH particle support with both mobile solids and wall boundaries.	87
13	Three cases of intersections between a spherical boundary and an SPH particle support.	88
14	Geometry of the case where a spherical boundary partly intersects an SPH particle support.	91
15	Example intersections of a circle and a triangular boundary element j , used in the calculation of γ_j and $\nabla\gamma_j$	99
16	Visualisation of the three-dimensional problem where an SPH fluid particle intersects a wall boundary element.	100
17	Calculation of γ for an SPH particle symmetrically approaching a corner.	106
18	Test problems of an SPH particle support intersecting multi-faceted complex boundaries.	107
19	Simulation of the 3D dam break test case using a basic semi-analytical boundary method implementation.	110
20	Simulation of the 3D dam break test case using the improved semi-analytical boundary method implementation.	113
21	Non-dimensional velocity magnitude of fluid particles settled into a hydrostatic box using the improved semi-analytical boundary method.	114

22	Relative error in the approximation $\nabla\gamma_i \approx \sum_w W_{iw}\mathbf{n}_w dS$ for a hydrostatic box of fluid.	116
23	Time evolution of the pressure in the 3D dam break test case using the improved semi-analytical boundary method implementation.	118
24	Time evolution of the free surface height in the 3D dam break test case using the improved semi-analytical boundary method implementation.	119
25	Comparison of the pressure at $P5$ in the 3D dam break for three different boundary method implementations.	121
26	Comparison of the free surface height in the 3D dam break for three different boundary methods.	122
27	Convergence study of the pressure in the 3D dam break test case using the improved semi-analytical boundary method implementation.	124
28	Convergence study of the free surface height in the 3D dam break test case using the improved semi-analytical boundary method implementation.	125
29	Schematic of a DEM solid collision between two solid particles [24].	132
30	Schematic demonstrating the main modes of bed load transport [4].	137
31	Visualisation of the simulation setup for periodic channel flow over a bed of mobile DEM bodies.	139
32	Trajectories of DEM bodies in a periodic channel flow over a bed of mobile DEM bodies, modelled with a shear lift force and without.	140
33	Visualisation of an SPH-DEM test problem where large DEM bodies are dropped into a box of SPH fluid. Results from both a fully exact “semi-analytical” porosity and a summed discrete porosity are shown.	142
34	Comparison of the total simulation time for the “semi-analytical” porosity calculation and the discrete summation porosity in a large scale SPH-DEM simulation.	144

1 Introduction

1.1 Coastal Hazards

Tsunamis, storm surges, and other large wave events have always presented threats to communities near coasts or large bodies of water. The largest of events threaten extensive socioeconomic damage and increased casualty rates. In the last two decades, events such as the 2004 Indian Ocean Tsunami, the 2010 Chilean Tsunami, and the 2011 Tohoku Tsunami have shone a light on the extensive destructive impact of tsunamis in an ever urbanising world. The 2011 event alone had approximately 20,000 casualties and was estimated to be the most costly disaster of all time when it happened [47]. Similarly, named hurricanes and superstorms have also garnered attention due to their extreme destructive potential. For instance, Hurricane Sandy in 2012 was estimated to have caused more than \$60 billion USD in damages [109], and Hurricane Ian more recently in 2022 was estimated to be the third-costliest weather disaster on record, with most of the damage caused by flooding due to storm surges [101] [102].

In response to these threats, engineers and policymakers have historically worked to improve the design of sea-defences to better protect communities and infrastructure. To develop appropriate defences, engineers typically rely on design codes, such as the ASCE codes in the USA, or international or national standards, such as the British Standards Institution. These include guidance on the design and construction of coastal structures to withstand different coastal hazards. Typical examples include sea walls, breakwaters, and flood defences [55] [75]. Ideally these codes are regularly updated based on the latest research to ensure that structures are designed to withstand the most severe coastal hazards. However, the effectiveness of these design codes is ultimately limited by the accuracy of the research and data on which they are based, and many design codes do lag behind the latest research. It has been noted that the European coastal engineering community appear to have limited awareness of forthcoming changes in standardisation [21], and a planned Eurocode on coastal structures expected in 2020 has still not arrived as of 2022 [59]. While designs for coastal structures do ultimately continue to improve over time, recent disasters and growing factors of modern global society pose new and pressing challenges in coastal hazard assessment.

The latest analysis on urban development suggests that in 2015 over 10% of the world's population lived less than 10 metres above sea level, with coastal population densities six times higher than the world average [29]. Further to an expanding presence of urban communities in areas at risk of coastal hazards, the effects of global urbanisation itself are also presenting increased potential risk. Rising sea levels owing to a changing climate [76], whether through the expansion of warmed oceans or the melting of the polar ice caps, will

increase the number of people affected by destructive storm surges and make flooding more frequent [123]. Frequency of tsunami-induced inundation may also drastically increase with only modest sea level rises [89]. The effect of higher ocean temperatures is two-fold, as this may also result in increased storm intensity and greater rainfall volumes [157, 128].

In addition to the direct risk on human lives within coastal communities, the accompanying infrastructure within these areas is also at risk. This carries wide-reaching implications when considering energy infrastructure, with potential widespread and long lasting power outages. A 2015 report by the Union of Concerned Scientists found that a large percentage of electrical substations in the U.S. were at risk from storm surges and noted the risk to the electricity infrastructure in the U.S. due to sea-level rise and coastal flooding [107]. Similar results were also found in a pilot study from the U.S. Department of Energy in 2014, which established the threat to energy infrastructure in the U.S due to sea-level rise and its associated coastal hazards [48]. Hurricane Maria in 2017 caused the largest blackout in U.S. history, and potentially the second largest blackout in the world [71].

The hazards and damage caused by coastal wave events¹ can occur in many ways. There is firstly the initial impact of a large wave, typically relevant to tsunamis and storm surges. These are characterised by high impact velocities and hydrodynamic forces, and the damage is often very extreme and caused over a short time scale. Tsunami waves will also mobilise substantial amounts of coastal sediment. Large waves can entrain large objects such as shipping containers, boats, cars, boulders, and other seafloor debris, which may in turn cause damage as they are dragged inshore. There are also less visible effects, such as sediment transport, seepage, scouring, and erosion. These may occur over longer time scales, at a wide range of velocities, and involve hydrostatic forces. Though less visible on the surface, these effects often involve increased complexity, and they are just as important when considering the hazards posed by large wave events [20, 46]. Tsunamis and storm-surges involve both short and long time scale effects, whilst flooding is primarily a long time scale hazard.

Offshore structures are also at risk of these kinds of hazards, particularly long timescale effects such as scour, as they are constantly exposed to water [154]. Offshore wind farms for example are particularly susceptible, with scouring affecting their fatigue life and stability [96, 103].

What is important to note in all these examples, is that they are all due to the interplay of the waves and surrounding solids. Evidently, an important aspect of coastal wave dynamics is the approach of the waves towards and onto the shore, typified by the interaction of the wave with different solids or structures around and within coastlines. In the case of offshore

¹Note here that the coastal wave events discussed will typically be in reference to any large destructive wave event near a coast or offshore structure, typically comprised of tsunamis, storm surges, and floods.

structures, the interaction is between the waves and the structures themselves or their solid defences or support. Any accurate studies of coastal wave events should be able to model and understand these interactions. These are complicated behaviours, but understanding fluid-solid coupling and fluid-structure interactions is a key step in producing more robust hazard assessment and protection models.

1.2 Fluid-Solid Coupling

Fluid-solid coupling and Fluid-Structure Interactions (FSIs) usually refer to similar problem types, with the important behaviour being that there is a flexible or deformable solid or structure that the fluid interacts with. FSI is often more specifically referred to in the context of fixed solids (structures), whilst fluid-solid coupling is more general and can allow for solid bodies that move within and around the flow.

Fluid-solid coupling and FSIs are of course relevant to coastal wave events, but not unique to this. They are in fact a common, though very complex engineering problem. Examples include aeronautical engineering - plane stresses under turbulence and heavy winds - and biomedical engineering - blood flow dynamics in arteries. Fluid-solid coupling is complex because it is a non-linear coupled problem. Non-linear in both the governing equations of the fluid and solid, and coupled from the combined interactions between the two phases. A flexible structure feels a forcing from the fluid, which in turn causes its movement or deformation, which then in turn affects the fluid flow, which then adjusts the forcing on the structure. This process repeats and thus describes the coupled state.

1.2.1 Coupling mechanisms in a coastal wave event

As already discussed, important fluid-solid coupling events and FSIs exhibit themselves in coastal wave events in a variety of ways. The key interactions of interest are:

- The impact of large waves on sea-defences - instantaneous hydrodynamic loading as well as sustained scouring.
- The erosion around the foundation of structures, either via scouring or liquefaction of foundation soil or sand.
- The shoaling process as large waves approach the shore and interact with the continental shelf and offshore boulders or structures.
- Entrained solids or debris within the fluid flow, which then alter the hydrodynamics and may impact with other structures and solids.

The consistent theme here is that coastal wave dynamics see waves interacting with a large range of structures/solids in many different processes, and in order to have a complete understanding of the fluid-solid coupling involved it is necessary to analyse all these different processes.

Whilst the focus here is on the mechanisms and behaviour of fluid-solid coupling, many of the studies relevant to disaster hazard assessment have focused on identifying these effects and their damage *a posteriori*, through the use of post-event field surveys. This is particularly relevant for both storm surges and tsunamis, where their unpredictability and extent of destruction make data collection during the events extremely difficult, and post-event modelling can only be done with data to guide the efforts. The aftermath of the events have often shown telltale signs of fluid-solid effects, such as debris impacts and scour, which are some of the key drivers of the extensive damage [46, 67, 60]. The identification of these effects in field investigations has shifted the focus on improving sea-defences away from just considering fluid forces towards the role that solid forces can play.

In the past, hazard assessments have typically foregone the damage of potential debris impacts. However, it is increasingly well understood that sea-defence failure mechanisms and large amounts of destruction are indeed due to debris effects [120, 124]. This includes short timescale effects such as impacts from large water-borne objects generating a large localised force, as well as longer timescale effects, such as damming, which leads to large areas experiencing increased hydrodynamic load. This has the potential to overturn the structure or dislodge its foundations. Smaller debris or sediment may also be included, as these can lead to increased effective fluid density and thus increased loading forces. Debris behaviour may be well-understood at a basic level, but debris loading in application to tsunami-like waves is relatively limited in scope — most of the literature focuses on single impact, single debris piece interactions.

Scour effects were identified as a dominant failure mechanism in sea dikes and seawalls in the 2011 Tohoku Tsunami [80]. This included the Tarō seawall which had previously withstood a tsunami in 1960. Failure mechanisms in this case typically involve destabilising the foundations of structures either during water inflow or draw-down and are usually due to scour of the foundation, tilting, or subsidence [133]. Whilst it is more often the case that scour is considered when assessing hazard risk to sea-defences, the myriad forms in which it can lead to structural instability present issues when trying to have a full understanding of the best way to combat it.

Field surveys are plentiful and particularly useful for identifying hazards post-event and collecting data points via observational studies. However, they must be supplemented with experimental, analytical, and numerical models to gain a more accurate hazard assessment

for future events.

Experimental studies of each of the fluid-solid interactions discussed are available, though vary in their application and use depending on the type of interaction considered. A concern when carrying out experimental models is scaling from laboratory scale to those encountered in real coastal wave events. This is partly because the smaller sediment substrate is modelled at full scale, while the waves are modelled at reduced scale. It has been noted that experiments involving grain-fluid mixtures, such as debris flows, may exhibit disproportionate viscous and pressure forcings relative to full-scale natural events [77]. Experiments should be made as large as possible to avoid this issue. Nonetheless, of all the fluid-solid coupled wave events, debris flows are some of the more straightforward to emulate, requiring only an appropriate mixture of fluids and solids and then allowing the wave to impact a structure or surface. Scheidl et al. [138] used a small-scale modelling approach to model debris-flow forces, relying on Froude number scaling to produce results applicable to natural debris-flow events. It is also sometimes possible to carry out these experiments on a relatively large scale, such as those presented in the debris flow experiment aggregation of Iverson et al. [78]. Work has also been done on how to properly apply small scale debris flow. The flows studied, however, are more relevant to landslides and floods in inland areas, where there is a larger amount of very small and permeable solids, such as mud, which can be easily suspended into the flow, leading to a blurring of boundaries between fluid and solid. These events are often well approximated as a single continuous non-Newtonian fluid flow rather than two distinct phases. This is less relevant to the current study, where the focus is on fluid-solid flows with more clearly defined boundaries between the fluid and solids. These would more appropriately be referred to as water-borne floating debris, creating a distinction from the discussed debris flows.

Water-borne debris flows, which have a clearer separation of fluid and solid phases, can be more difficult to set up with large scale experiments. They usually involve larger solid objects and much greater forces, thus requiring an experimental set up that is able to handle such dynamics. For this reason, experimental studies are usually done on a lab scale, and primarily used as validation for numerical or analytical models.

Smaller scale sediment transport problems such scouring or liquefaction have benefited from experimental studies. In 2003, Tonkin et al. created scale-model experiments to investigate the then-relatively understudied transient scouring effect due to tsunamis and storm surges [155]. They highlighted the importance of pore pressure gradients for different scour mechanisms. This work has been expanded by McGovern et al. who studied onshore tsunami scour for longer inundation time-scales [50]. Larsen et al. carried out lab scale experiments to study tsunami-induced scour around a monopile foundation such as those found in offshore

wind turbines [87] and were able to correspond their results to real-life tsunamis. Chen et al. have carried out many studies on the susceptibility coastal structures to tsunami-induced scour [27], both identifying the extent to which scour occurs, as well as considering different breakwater configurations to limit the effects. It was also found that typical sea-defence setups such as submerged breakwaters are ineffective at reducing scour.

Larger scale water-borne debris flows, involving debris transport and impacts, can also be carried out on a laboratory scale, but with more limited configurations due to the much larger forces involved and potentially greater concerns around the effects of scaling. Initial work on debris transport was carried out by Imamura in 2008 [74], who ran basic tests in a lab flume of the transport of a single boulder in a dam break flow and used this to develop a basic boulder transport model. Yao et al. [166] ran a preliminary investigation into the motion of debris in a tsunami-like solitary wave flume, using scaled down models of coastal houses. They noted the difficulty of using experimental results as a proxy for the natural processes due to the distorted scalings involved in an experimental set-up of debris motion. Nouri, Nistor, and Palermo [125] produced one of the first experimental studies on the impact of floating debris in a tsunami wave in 2010. They used wooden logs in a dam break flow to produce preliminary findings, such as the potential for debris bounceback leading to multiple impacts. Shafiei et al. [142] were able to attach an accelerometer to a 1:20 scale debris downstream of a rapid release of water (akin to a dam break flow) and measured the forces of the subsequent debris impact against a structure. They found that the added mass of the entrained water due to the debris motion was an important term in the impact force calculation. For simulating tsunami-borne debris impacts, scaled down shipping containers are often used as the debris of choice. Derschum and Nistor et al. [39] ran experiments with a 1:40 scale shipping container in a dam break flow and examined the influence of transient flow conditions on the debris impact. They found that non-linear interactions of the flow and debris leads to complex movement patterns of the debris before and after impact, such as multiple impacts, confirming earlier findings from Nouri et al. [125]. Riggs and Cox et al. [82] ran full scale experiments in air, and a 1:5 scale shipping container experiment in water. They compared the in-air and in-water results with each other and with analytical predictions. In-water impacts were observed to have an increase of $\sim 17\%$ in peak impact forces and $\sim 20\%$ in impact duration.

In all these experiments, in order to simulate the behaviour of large waves in a laboratory, it is usually the case that the waves are generated in a flume via a mechanical wave maker that can generate tsunami-like wave motion, or a dam break style wave where a body of fluid is released from rest. These two methods for simulating large coastal waves, in particular the dam break, will be frequently employed in the results presented.

Within many of these experimental studies, basic analytical models are often referred to as a benchmark for results. These are generally based on basic fluid-dynamics equations with many simplifications to calculate flow velocities, pore pressures, and impact forces.

Work on analytical predictions for tsunami scour and sedimentation processes are very limited, and most of the models are heuristic and guided by basic lab experiments. Yeh has addressed the analytical study of sediment response to tsunami loading [170, 169, 168], with models based around estimating pore-pressure gradients during the tsunami run-up and drawdown process and the resulting liquefaction and instabilities. These models are not often found elsewhere in literature.

For water-borne floating debris transport and impacts there exists slightly more well-developed analytical models. Imamura's work in 2008 [74] produced an effective basic model to estimate and better understand debris transport processes. Riggs and Cox created an analytical model based on a solid-only impacts and stress wave propagation in the debris [134]. Their equations based on simple wave propagation provided adequate estimates for peak impact force but less so for impact duration. Nistor et al.'s review of tsunami-driven debris loads contains a discussion of the typical analytical models used in estimating debris motion such as transport, impact, and damming [124]. They conclude that rigorous, high-resolution studies are required to provide more detailed, and tailored benchmarks, as well as to assess more complicated situations such as multiple impact events. Shekhar et al. recently produced work attempting to address this by conducting experimental studies involving a debris field, and studying more complicated effects such as the impact of debris orientation and damming [144].

Whilst analytical models can be effective in producing basic order of magnitude results, they usually are unable to account for more complex phenomena involving coupling between the fluid and solid and three-dimensional effects. Numerical models are therefore often seen as the next step in producing high quality data at a reasonable expense.

Sugawara et al. wrote a review of the current understanding of numerical modelling of tsunami sediment and boulder transport in 2014 [150], where they discussed the use of forward models which address the prediction of sediment transport and deposits due to wave behaviour. They also described inverse models which estimate wave behaviour based on observed sediment behaviour. Many of the models discussed in this review are only two-dimensional and use parameterised models of the solids based on simple equations. Larsen et al. produced a numerical assessment for tsunami-induced scour [88], leveraging finite-volume CFD (computational fluid dynamics) simulations with previous scour knowledge in order to develop a scour prediction methodology. The numerical experiments were limited

by computational expense and focused on only individual monopile foundations at a lab scale. Son et al. [147] created a fully coupled sediment transport model applicable to tsunami waves through use of the finite-volume method to solve a coupled Boussinesq model, sediment transport model, and morphodynamic models. Results were only validated and produced on one and two dimensional test cases. Further to the discussion on analytical models, Nistor et al.'s review of tsunami-driven debris loads [124] also contains a review of the available numerical models. The majority of effective models in this area were found to be particle based methods, such as the Moving Particle Semi-implicit method, Smoothed Particle Hydrodynamics (SPH), and the Discrete Element Method (DEM). Nistor notes that SPH is being increasingly used as a tool to simulate free-surface flows with multi-body dynamics. These will be discussed in more detail in the following section.

1.3 Numerical modelling for Fluid-Solid Coupling

As discussed above, fluid-solid coupled problems can be difficult to accurately model. Problems are often too complex to be solved analytically, and lab experiments are limited in scale or scope [70]. Numerical solutions have therefore emerged as a promising method to investigate these complex interactions.

However, the complexities of the problems must still be accounted for in numerical models. The problems usually involve multiphysics, with the fluid and solid having separate physics and behaviours and a large range of scales. The system must also be coupled with interlinked equations of motion that must be updated appropriately. The boundaries between the fluid and solid are transient interfaces which present discontinuities in state variables. Solids may range from being large and stationary to very small and mobile, with different mechanics affecting each, and these should be able to be concurrently assessed. If a free surface is involved, as is often the case in coastal wave problems, then this is yet another interface which may rapidly change in time.

Conventional mesh-based methods are the most common form of computational fluid dynamics implementations, but tend to struggle with some of the aforementioned complexities. A mesh-based method is one whose simulation nodes fall on an underlying, usually structured and fixed, mesh. Examples include the finite difference, finite volume, and finite element methods (FEM). The mesh allows simple connectivity to surrounding nodes and discretisation of continuum equations, but causes problems when handling interfaces or discontinuities that do not coincide with the mesh lines [165]. The volume-of-fluid (VOF) method is commonly applied to mesh-based methods in order to model a free surface and

requires a further advection scheme to be solved alongside the standard flow solving algorithm.

Meshless methods offer the advantage of flexibly defined simulation nodes, which is ideal for fluid-solid problems with transient boundaries and discontinuities. They can be formulated in a Lagrangian description, meaning that the nodes (*“particles”*) of the simulation can move with the fluid flow and any other movement in the system. This offers many advantages. For example, naturally tracking the interfaces within the system - particularly important when there is rapid evolution of the free surface and large movements of the fluid-solid interface. Meshless Lagrangian methods are often referred to as “particle methods”, since they essentially model the system as particles that are free to move with the system. Each particle is a node which carries information relevant to the physics and calculations, and is an integration point for the method. Fixed mesh methods are akin to fixed buoys in a fluid which measure the flow characteristics at given points, whilst particle methods are akin to free buoys which flow with the fluid whilst measuring the flow characteristics. There are a variety of particle methods available, such as the previously mentioned Discrete Element Method (DEM), Smoothed Particle Hydrodynamics (SPH), meshless local Petrov-Galerkin (MLPG) method, and finite volume particle method (FVPM), just to name a few. DEM is used for solid mechanics and calculates particle overlap and rotation using rigid body models to calculate interactions. SPH uses a smoothing kernel to approximate field variables and their derivatives. MLPG solves a local weak form of the governing equations at each particle using a moving least squares approximation. FVPM replaces control volumes in the conventional finite volume method with overlapping particles whose interactions are analogous to the fluxes between volumes in the finite volume method. Each of these offer their own advantages and disadvantages, but all retain the advantages of particle based methods in general. The SPH and DEM methods are the most commonly applied to coastal dynamics problems, and often coupled, are the focus of the work here.

In addition to mesh-based and meshless methods, there also exist “hybrid”² approaches that combine aspects of both. Hybrid approaches allow for the simulation of large deformations and movements while maintaining some advantages of a fixed mesh, such as accurate representation of the geometry and straightforward computation of the material properties. One such approach is the particle finite element method (PFEM), which uses a finite element mesh to discretise the domain but allows the nodes of the mesh to move according to Lagrangian motion and remeshes the nodes when it is necessary. Another hybrid method is the material point method (MPM), which combines Lagrangian “material points” with a

²These are often just categorised as meshless methods since they avoid the typical drawbacks of mesh-based methods

fixed background mesh onto which their quantities are extrapolated in order to calculate derivative terms. Due to having to maintain both a connected mesh and a set of flexible nodes, with appropriate parameters for both, these hybrid methods encounter limitations when it comes to computational cost and efficient implementation. Nonetheless they have been successfully applied to a wide range of fluid-solid coupled problems, including FSIs [106], and can offer unique advantages over both mesh-based and meshless methods in some cases.

1.3.1 Discrete Element Method

The modern form of DEM was first proposed by Cundall and Strack in 1979, who developed a numerical model for simulating the interactions of discrete granular particles. The ability to model discrete individual particles lends itself to granular media such as grains, rocks, and sand, rather than continuous media like fluids. As such, DEM is often used in geological problems, for rock and soil mechanics.

Classical DEM works by first calculating the contact points of all bodies and the forces resulting from these contacts, governed by what are called the constitutive equations. This includes friction, stress and strain, and collisions. During collisions the bodies will undergo a mathematical deformation which is used to calculate the contact forces in what is known as a spring-dashpot model [121]. This will be discussed in further detail in Chapter 6.

The contact forces, and any other body forces such as gravity or the forces from any coupled method, are then used in the equations of motion to calculate the acceleration. The equations of motion for DEM are based on Newton's Third Law and the Newton-Euler equations for rigid-body dynamics. This methodology means DEM can be easily coupled with other numerical models. Fluid-solid numerical models are often developed in the form of a coupling between DEM and another numerical method for modelling the fluid, commonly known as CFD-DEM (Computational Fluid Dynamics - Discrete Element Method) methods, with specific examples such as FEM-DEM, SPH-DEM and MPS-DEM.

The focus here will be on its coupling with the SPH method.

1.3.2 Smoothed Particle Hydrodynamics

SPH was first developed independently by both Lucy [95] and Gingold and Monaghan [62] in 1977. The main objective of SPH is to reconstruct a continuum from a cloud of discrete particles. It was first applied to solve strongly compressible astrophysical problems, with the main attractions being its ability to exactly model advection, particularly at high velocities, to naturally track interfaces between different fluids or phases, to implicitly handle variable

resolution, and to bridge the gap between continuum and fragmentation [114]. All of these advantages come from the particle description of the system, where different fluids or phases are assigned a different set of particles carrying their own unique properties. The movement of the particles with the flow ensures that integration nodes are always in the high density areas, ensuring resolution where it is required. Different interacting fluids or phases are always exactly tracked. Advection is treated exactly, as it is a lagrangian method where all of the fluid properties are transported with the integration points.

These are all general behaviours and advantages of particle methods, though SPH is itself a unique technique. In SPH the particles represent a region of space, possess material properties, and interact with each other within the range of a defined “*smoothing length*”. This smoothing length describes the range over which a particle’s properties are “smoothed”. By smoothing the properties of discrete points in this way, it allows for the reconstruction of a continuous field. Contributions from the local particles are summed to find the continuous field. The method will be described in more detail in Chapter 2.

SPH may be the most well-developed modern meshfree method. In 2005, SPHERIC (originally “SPH European Research Interest Community”, now “SPH rEsearch and engineeRing International Community”) was developed as a central hub to gather SPH researchers and users and streamline the work and developments of the method [149]. This, along with the merits of the method, has helped propel SPH to its position as one of the most well-developed and widely used modern meshfree methods. The most recent review on the method by Lind et al. [91] noted that since 2016, six review papers on SPH have been published.

SPHERIC identifies “Grand Challenges” [160] as the major areas of development for SPH needed to bring it to the forefront of numerical modelling alongside the more well-established and widely used mesh-based methods. Of relevance to this work are the grand challenges of Boundary Conditions and Coupling to other methods. Both of these issues will be discussed within this work.

Alongside a discussion on the Grand Challenges, SPHERIC presents relevant literature, key validation tests and results for the method, and a collation of all the available open-source codes which implement the method. One of the attractions of SPH is this wide and active community, which is continuously developing and adding to the method and providing open source software.

SPH has been applied to achieve a variety of results and validations, particularly in the realm of coastal engineering. Early work in the field from Dalrymple applied SPH to water waves impinging on structures and walls [36]. Dalrymple then later worked with Rogers applying the method to breaking waves on beaches and wave-structure interactions based

on a dam break around a fixed structure [37]. Barreiro et al. applied the method using the *DualSPHysics* open-source code to solve problems of wave propagation in coastal areas and wave impact on structures with good comparison to experimental data [14]. Trimulyono et al. validated SPH against experimental wave generation and propagation data [158]. Vacondio et al. modelled the Vajont rockslide and the wave it generated with a full 3D implementation and achieved satisfactory comparisons to the field data [159]. Canelas et al. have coupled the *DualSPHysics* open-source code with a DEM solver [25] and a separate physics library *Project Chrono* [23] in order to simulate fluid-structure interactions. These methods are able to model fluid interactions with large arbitrary structures such as wave-energy converters or tidal turbines, but do not address other smaller scale interactions such as scour or sedimentation. More recently, SPH has evolved into a more complete tool for coastal engineering problems, investigating and producing validated results on subjects such as the forces exerted on harbour protections and the run-up of tsunami-like waves on beaches [44]. SPH has also been successfully applied to industrial problems [141]. It is often the case, as with most computational fluid dynamics (CFD) solvers, that certain algorithms, accuracy improvement methods, and other parameters of the model, have to be tweaked to account for the nuances of a new problem with different dominating regimes and physics. This means there remains a lot of potential for new developments and problem applications.

Recently at Imperial College London, Professor Stephen Neethling has developed an in-house fully fluid-solid coupled SPH code [122]. This has provided an ideal opportunity to expand the method within the Department of Earth Science and Engineering, and apply it to further problems of interest for which SPH may provide advantages that the traditional methods have struggled with - namely fluid-solid coupling. Of particular interest at this point in time are problems involving mobile solids, which represents an even more numerically understudied field than standard FSIs.

1.4 Outline of Thesis

This work will first introduce the SPH method and its conceptual framework, as well as a full mathematical formulation of the numerical techniques employed in the current SPH model with relevant results and discussion. This will mostly include a discussion of the standard techniques in SPH and some of their implications which are often understudied in the literature, but important when considering how to contextualise accuracy and errors in an SPH simulation. This is relevant, as the presence of solid boundaries in fluid-solid coupling is particularly problematic due to the manner in which it deteriorates the accuracy of the method. Newly developed techniques to overcome these problems will also be presented.

The computational implementation and considerations will be addressed at each step.

The first contextual focus will be the handling of fixed boundaries within SPH. The different methods for modelling boundaries in SPH will be discussed alongside additional developments studied in this work. Particularly of note, a new complete formulation of the so-called semi-analytical boundary method will be presented, with a focus on the calculation of the complex normalisation terms present in the method, as well as how this can be implemented in any existing SPH code. The aim is to present an updated semi-analytical boundary method which can be simply implemented into existing codebases, alongside relevant context on the method to provide a more complete assessment of how it can be best leveraged and employed in SPH models.

The coupling of SPH and DEM for modelling mobile boundaries will then be briefly discussed. A distinction will be made between large and small-DEM, whereby the modelling of solid DEM particles smaller than the fluid resolution will be addressed. This provides the framework for a multi-scale coupled SPH-DEM code. There will be a discussion on the current improvements and results found for small-DEM, leveraging techniques from the previously discussed new semi-analytical boundary method. Future work will involve validations and basic demonstrations on this framework.

Finally, the work will be concluded by consolidating the results and discussing shortcomings and potential further improvements of the current work. Much of the validations and demonstrations of the method will be left for future work.

The aim is to provide in-depth discussion on how solid boundaries, both fixed and mobile, can be modelled within SPH, and in doing so present new work on the semi-analytical boundary method and the coupling of multi-scale DEM with SPH. The discussion and final model should offer good insight and improvements into the accurate modelling of fluid-solid coupling, particularly in the context of coastal hazards.

2 Smoothed Particle Hydrodynamics Method

The task of accurately modelling solid boundaries in SPH requires a clear understanding of the method and its mathematical underpinning. SPH is already limited in its potential accuracy (as will be seen in Sec. 2.3) and solid boundaries present even further difficulties due to how their presence complicates the governing equations and further limits the accuracy that can be achieved. A mathematical framework of the method is required to understand the limitations on the accuracy due to solid boundaries and also better understand ways in which these inaccuracies can be counteracted.

This section will start by discussing the core method (already briefly discussed) and conceptual framework, with an understanding of the technique and equations from a more heuristic point of view. A mathematical formulation of the SPH method will then be presented, including a discussion on its accuracy and where the limitations reside. This will help contextualise the difficulty of modelling solid boundaries further in the work and also provide necessary background for when the semi-analytical boundary method is introduced and how it tackles the issue of solid boundaries. The numerical techniques which work to extend capabilities or improve accuracies of SPH will then be discussed, aiming to cover at a base level the core tenets that all SPH implementations have to consider. It should be noted that new and improved numerical techniques are constantly being developed, of which a selection which are relevant to coastal dynamics problems will be discussed. During this discussion, improvements on previous methods will also be presented. Finally, the computational implementation of the developed code will be discussed. In total, this serves to act as a review of the current SPH implementation and the aspects which are key to improving the modelling of fluid-solid coupled simulations and the handling of solid boundaries.

There exist many extensive reviews of the SPH method [114, 115, 32, 130, 167, 162, 92]. Particularly of note is one of the first and only books published on the SPH method by Violeau [161]. This contains probably the most complete review of the method. Though it is now outdated in some sections and missing newer techniques, the mathematical formulation it discusses still remains very relevant and contains a complete discussion of much of the mathematical foundation of SPH which is often missed in newer literature. The discussions on the method by Price [131, 130] were particularly influential in guiding understanding of some of the less discussed nuances to SPH and why the method exhibits certain behaviours. The cited SPH reviews, as well as the numerous discussions in the introductions of shorter papers, have provided a general influence and inspiration for many of the discussions in this section.

2.1 Conceptual Framework

Any CFD code works by discretising the continuum fluid dynamics equations onto a set of discrete points. Conventional mesh-based solvers discretise onto a fixed grid - the mesh - whilst meshless methods discretise onto “particles”. In this sense, therefore, the particles are the nodes, or integration points, of the system - just like a grid point in a mesh-based method.

As described in the Introduction (Chapter 1), the key caveat is that the particles are allowed to move according to the material properties at that point. This means that not only do the particles represent the computational grid points, but also, if assigned properties and equations properly, the fluid itself too. As the particles move, this represents the fluid moving. The integration points flow with the fluid and naturally track its movement, and this is why the methods are so effective at handling interfaces and advection. This, however, describes any Lagrangian particle method, whereas SPH also has a unique “smoothing” process and other considerations.

The SPH scheme will now be fully addressed, including its mathematical formulation along with a discussion of some recent developments to the techniques and other ways the method can be optimised.

When creating a CFD method, the question to first propose is, for any discrete set of points, how is it possible to approximate the continuum equations? This essentially becomes an interpolation problem. We wish to interpolate data which is only defined at specific points to any point within the domain. We hence wish to create a “smooth” and continuous field from a set of discrete points.

We will first approach this in the 1D formulation to allow us to visualise the process more easily. Instead of representing each particle as a single point in space, we apply a smoothing function, and thus smooth its values over a distance around it. An initial and the most classic example of a smoothing function is a Gaussian, which retains a peak at the centre and gradually and symmetrically drops off to zero further from the centre. Applying this smoothing function to a single point - representative of a Dirac delta function - we reproduce a smoothed representation of the value. By stacking - or summing - these smoothed representations, we reconstruct a continuous function over the domain. In this way, a value can be reproduced at any point by a weighted summation over the nearby particles. This can be seen in Fig. 1. This illustrates in 1D how discrete values are smoothed and then combined to reconstruct a continuous and smooth field.

The smoothing function, which in this case was a Gaussian, is referred to as the *kernel function*, and the characteristic length which controls the scale of smoothing for each particle is the *smoothing length*, h . These are two key aspects of the SPH method.

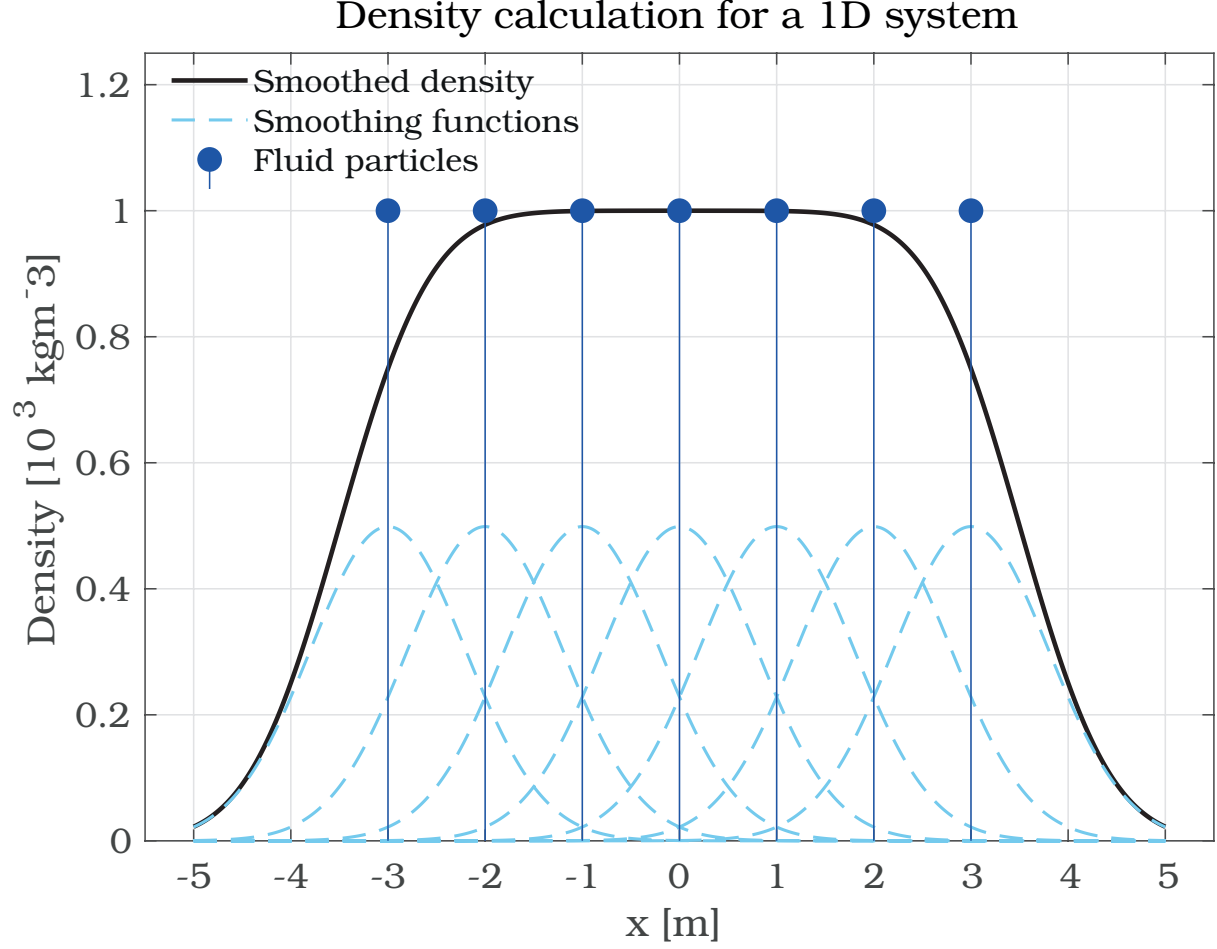


Figure 1: The process of smoothing discrete values to a continuous field in 1D. The blue points and vertical lines represent discrete density values of fluid particles in 1D. The dashed cyan lines represent a smoothing function - “kernel” - applied to the discrete points to produce a local smoothed contribution. The black line represents the summation of the smoothed cyan lines across the domain producing the final smoothed density field. This process therefore represents the SPH approximation of discrete points to a continuous field.

2.2 Mathematical Formulation

The next step is to formulate this mathematically in a full 3D context. We will be smoothing known values $A(\mathbf{r}')$ at positions $\mathbf{r}'$, with a kernel function $W(\mathbf{r} - \mathbf{r}', h)$ of radius h centered at our desired position $\mathbf{r}$. These smoothed values are then summed over the domain of local points - $\int_{\Omega} d\mathbf{r}'$ - to find a reconstructed value $A_I(\mathbf{r})$ at the desired position. The integral $\int_{\Omega} d\mathbf{r}'$ covers the full domain in as many dimensions as necessary, and may for example be seen as an area integral in two dimensions or a volume integral in three dimensions.

$$A_I(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}'. \quad (1)$$

$A_I(\mathbf{r})$ represents the continuous integral interpolation value at the position $\mathbf{r}$ and Ω is the function domain. The formulation is effectively an extension of the integral representation of a function defined using the Dirac delta function: $A(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}') \delta(\mathbf{r} - \mathbf{r}') d\mathbf{r}'$. The SPH approximation is produced by replacing the Dirac delta function with the regular and more practical kernel function W . The specific form and requirements of the kernel function will be discussed in Sec. 2.3.

Noting that the system consists of discrete particles, it is necessary to approximate this integral $A_I(\mathbf{r})$ as a discrete sum $A_D(\mathbf{r})$. This can be done as a Riemann sum over all the local particles as

$$A_D(\mathbf{r}) \equiv A_i = \sum_j A_j W(\mathbf{r}_i - \mathbf{r}_j, h) V_j. \quad (2)$$

We have defined A_i as the value at particle index i , and index j as the indices of the neighbouring particles which are included in the sum over the local domain. In this way, going from the continuous approximation to the discrete, we have $\mathbf{r} \rightarrow \mathbf{r}_i$ and $\mathbf{r}' \rightarrow \mathbf{r}_j$. V_j is introduced to represent the volume of a neighbouring particle and replace $d\mathbf{r}'$ in the integral formulation. We can also rewrite $V_j = m_j/\rho_j$, where m_j and ρ_j are the mass and density of the particle j respectively.

Hereon, throughout this work, $A(\mathbf{r})$ will be used equivalently to the SPH approximation $A_D(\mathbf{r})$. This is extended to all SPH approximations discussed. Of course, it should be noted the use of the $=$ sign is purely a notational convenience, and in fact SPH approximations are all a discrete approximation to a continuous approximation. The implications of this will be discussed in Sec. 2.3.

The smoothing length h may be chosen to be a variable or a constant. A variable smoothing length introduces additional complexities in the SPH formulation and implementation. For the astrophysical simulations that motivated SPH this was a necessity, due to the large range of length scales involved in the problems. For coastal dynamics problems, however, this is not the case, and so most SPH models in this domain assume a constant smoothing length. We will do the same here, and so omit the h from the notation for W .

When we wish to refer to the value of a function at a particle we will use A_i , where i then refers to the index of the desired particle, and we thus also introduce $\mathbf{r}_i$ as the position of the desired particle.

2.3 Kernel function and accuracy of the method

The smoothing kernel thus far discussed is important to consider, as it plays a vital role in the accuracy and overall limitations of SPH. Generally, all of the numerical techniques and boundary methods that will be discussed within this work aim to maximise the accuracy of the method, and so work to counteract the inherent inaccuracies which naturally arise from the smoothing and discretisation process. Understanding these inaccuracies allows better understanding of how the accuracy of the method may or may not be improved and theoretical limits to any numerical technique or boundary method which we will introduce. We will here address the role of the kernel function, how this mediates the accuracy of the method, and other sources of accuracy limitations. Regarding truncation errors due to the method itself, Quinlan et al. [132] produced the first truncation error analysis for SPH in 2006 and remains an important resource when considering error behaviours in SPH. The accuracy limitations posed by the presence of solid boundaries will be discussed later in Chapter 3.

So far only the Gaussian function as an example of the smoothing kernel has been mentioned, but in fact any function with the desired properties may be used.

The kernel should firstly tend to the dirac delta function as the length scale, h , tends to zero

$$\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r}) \quad (3)$$

This ensures that using the integral representation, the original function can be reproduced exactly as resolution tends high enough. Further constraints on the kernel function can be assessed by looking at the accuracy of the approximation (1).

To do this, we first take a Taylor-series expansion of $A(\mathbf{r}')$ about the point $\mathbf{r}$ giving

$$A(\mathbf{r}', t) = A(\mathbf{r}, t) - (\mathbf{r} - \mathbf{r}') \cdot \frac{\partial A}{\partial \mathbf{r}} + \frac{1}{2} (\mathbf{r} - \mathbf{r}')^T \frac{\partial^2 A}{\partial \mathbf{r}^2} (\mathbf{r} - \mathbf{r}') + \mathcal{O}(|\mathbf{r} - \mathbf{r}'|^3) \quad (4)$$

where the derivatives are evaluated at position $\mathbf{r}$. The order of accuracy $\mathcal{O}(|\mathbf{r} - \mathbf{r}'|)$ is synonymous with the length scale of the particles i.e. the resolution of the simulation. Substituting this equation into Eq. (1) gives

$$\begin{aligned} A_I(\mathbf{r}) = A(\mathbf{r}) \int_{\Omega} W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' - \frac{\partial A}{\partial \mathbf{r}} \cdot \int_{\Omega_0} (\mathbf{r} - \mathbf{r}') W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' \\ + \frac{1}{2} \frac{\partial^2 A}{\partial \mathbf{r}^2} : \int_{\Omega} (\mathbf{r} - \mathbf{r}') \otimes (\mathbf{r} - \mathbf{r}') W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' + \int_{\Omega} \mathcal{O}(|\mathbf{r} - \mathbf{r}'|^3) W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' \end{aligned} \quad (5)$$

where the third term on the right-hand side has been rewritten with the double dot product “:” and tensor product, noting that $\frac{\partial^2 A}{\partial \mathbf{r}^2}$ is a second rank tensor and thus

$$(\mathbf{r} - \mathbf{r}')^T \frac{\partial^2 A}{\partial \mathbf{r}^2} (\mathbf{r} - \mathbf{r}') \equiv (\mathbf{r} - \mathbf{r}') \cdot \frac{\partial^2 A}{\partial \mathbf{r}^2} (\mathbf{r} - \mathbf{r}') = \frac{\partial^2 A}{\partial \mathbf{r}^2} : (\mathbf{r} - \mathbf{r}') \otimes (\mathbf{r} - \mathbf{r}'). \quad (6)$$

This term is fully expanded in Appendix A.

In order for the approximation $A_I(\mathbf{r})$ (Eq. (5)) to have a second-order accuracy to the real function value, that is $A_I(\mathbf{r}) = A(\mathbf{r}) + \mathcal{O}(|\mathbf{r} - \mathbf{r}'|^2)$, it requires two conditions:

$$\begin{aligned} \int_{\Omega} W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' &= 1 \\ \int_{\Omega} (\mathbf{r} - \mathbf{r}') W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' &= \mathbf{0} \end{aligned} \quad (7)$$

The first condition is effectively a normalisation condition which stipulates that the kernel should integrate to 1 over its domain, or equivalently its zeroth-order moment should equal 1. The second condition says that the first-order moment should equal 0, and this can be satisfied if we take the kernel to be an even function, that is $W(\mathbf{r}) = W(-\mathbf{r})$. These are the two minimum requirements on the kernel function in order to ensure suitable accuracy of the method. The normalisation condition also allows us to see that the second to last term in Eq. (5) is second order in $|\mathbf{r} - \mathbf{r}'|$, thus confirming that with the two conditions (7), $A_I(\mathbf{r}) = A(\mathbf{r}) + \mathcal{O}(|\mathbf{r} - \mathbf{r}'|^2)$. This is often written as $\mathcal{O}(h^2)$, since for kernels with compact support the smoothing length is the same order as the largest interacting particle distance and thus is a proxy for $(|\mathbf{r} - \mathbf{r}'|)$ and the simulation resolution.

It is even possible to increase the accuracy further to third order by setting the second-order moment of the kernel to 0, however this would require the kernel to become negative at some points which is problematic from a physical point of view and so is not usually done. A further analysis of the kernel functions and its implications for accuracy of the method can be found in both the book by Violeau [161] and other literature [92, 93, 51]. There are other desirable properties of the smoothing kernel, such that it should be sufficiently smooth, twice continuously differentiable, and monotonically decreasing. These conditions help produce stable and accurate results, and monotonically decreasing can be understood in a physical sense as ensuring that the closer a particle is the more it will influence the result.

Another condition that is almost always placed on the kernel function which has computational rather than mathematical motivation, is that it should have a finite support domain. This is to say

$$W(\mathbf{r} - \mathbf{r}', h) = 0 \quad \text{for } |\mathbf{r} - \mathbf{r}'| > \kappa h \quad (8)$$

where κ is some scaling factor, usually selected as 2, and h is the smoothing length of the kernel. This condition is also known as compact support, and implies that the kernel function

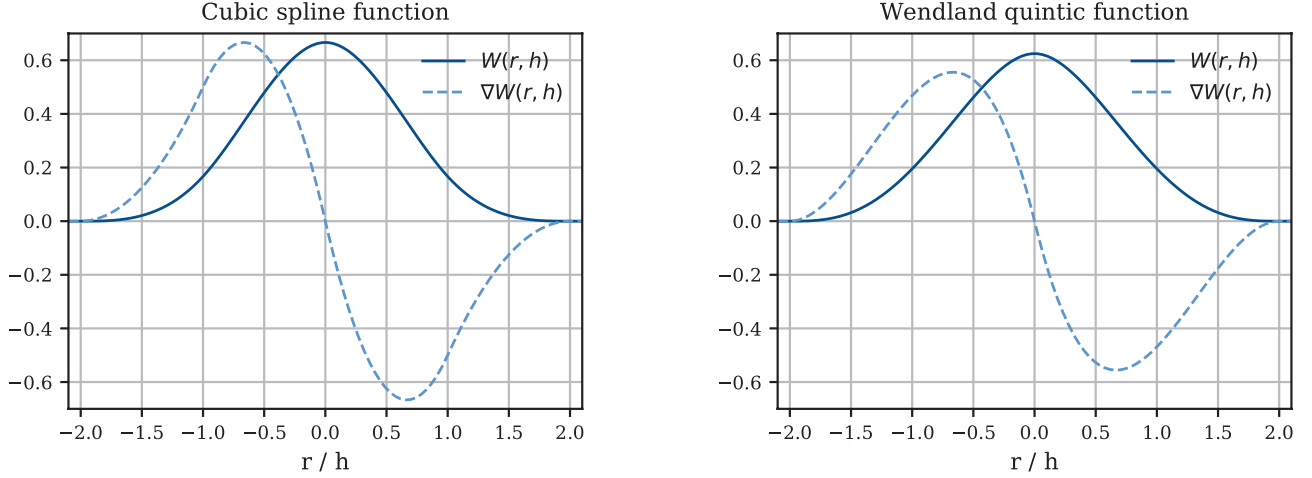


Figure 2: Plots of the Cubic spline and Quintic Wendland functions and their derivatives. The smoothing length of each function is h and the compact support is $2h$

is locally limited. This is a computational requirement to ensure that particle interactions do not have to be calculated over the entire domain. A finite support ensures that only particle interactions within a certain range - $2h$ - need to be calculated. This means, for example, that the sum in Eq. (2) is only over particles within $2h$ of the given position. This stops simulations becoming prohibitively slow by calculating the interactions between every single particle in the simulation at every single timestep. Since the kernel functions rapidly drop with distance, particles that are closest are the ones which influence the calculations most strongly, and so a finite support domain is chosen as a balance between accuracy and speed. For example, a Gaussian function has infinite support domain (it tends to 0 at infinity), but its value drops to 0.013% at just 4 standard deviations, so any interactions beyond that point are largely negligible.

As noted, though the Gaussian function is a natural candidate for the smoothing kernel, it has infinite support domain and so is not ideal. Other kernel functions have been developed, and are now commonly used in SPH, such as the Cubic Spline [118], which offers good accuracy and running times, and the Quintic Wendland [163] which is higher order and so offers better accuracy, but at a slightly increased computational cost. The plots of these two smoothing kernels and their derivatives are shown in Fig. 2. A more complete review of available kernel functions can be found in the review by Liu and Liu [92]. The kernel function used in the SPH implementation presented in this work is the Quintic Wendland.

So far we have only addressed the accuracy of the method in terms of the original integral approximation using the kernel function, but there is of course the further discrete approximation as $A_D(\mathbf{r})$ (Eq. (2)) which also entails its own errors. The accuracy of the discrete

approximation is harder to analyse since it depends on the particle spacings δr , which can freely change during the simulation. Violeau presents a discussion of the issue [161] and estimates that totalling the integral and discrete approximation for the Quintic Wendland kernel produces an accuracy of the form

$$A_D(\mathbf{r}) = A(\mathbf{r}) + \mathcal{O}(h^2) + \mathcal{O}\left[\left(\frac{\delta r}{h}\right)^6\right] \quad (9)$$

The implication of this is that while increasing h widens the integration domain and reduces the accuracy of the integral estimation at the local point, it also allows more particles to be included in the summation which reduces the statistical inaccuracy of the discrete estimation. These two effects are in opposition, but ultimately the dominating effect is that owing to the integral approximation of $\mathcal{O}(h^2)$. This does however clarify that it is not enough to reduce the smoothing length excessively, as it is important that the ratio $\delta r/h$ does not increase too much. This is analogous to the inverse of the number of particles in a smoothing length, and ensures that the discrete sum approximation maintains an adequate number of contributing neighbouring particles. This is usually handled by setting the ratio $\delta r/h$ as a constant in the range of $0.6 \sim 0.8$, which ensures that the error owing to the discrete estimation remains fixed at any resolution. Additionally, this clarifies the idea that maintaining a regular particle distribution is very important in obtaining accurate results, something that is often addressed throughout SPH implementations.

This analysis is carried out more completely in Quinlan et al. [132], which contains a full analysis of some of the ideas discussed here concerning the roles of both h and δr in mediating the error of SPH.

Only the accuracy of the direct SPH estimate has been assessed here, but the same analysis can be applied to the SPH estimate of the gradient of a quantity and an equivalent result is found [130], meaning this error analysis is valid throughout an SPH scheme.

2.4 Derivatives in SPH

Equation (2) is how a quantity is directly estimated in SPH, but of course in any fluid dynamics solver it is necessary to also evaluate the gradient, divergence, and any other derivative quantities. Both Violeau [161] and Price [130] discuss in good detail the nuances and intricacies that are often missed when it comes to analysing gradient operators in SPH.

We can first start by applying the integral interpolant approximation to the gradient of a variable.

$$\nabla_{\mathbf{r}} A(\mathbf{r})_I = \int_{\Omega} \nabla_{\mathbf{r}'} A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|) d\mathbf{r}' \quad (10)$$

We will hereon drop the subscript I for the gradient approximation noting that all calculated quantities going forward are effectively approximations. Applying integration by parts

$$\nabla_{\mathbf{r}} A(\mathbf{r}) = \int_{\Omega} \nabla_{\mathbf{r}'} [A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|)] d\mathbf{r}' - \int_{\Omega} A(\mathbf{r}') \nabla_{\mathbf{r}'} W(|\mathbf{r} - \mathbf{r}'|) d\mathbf{r}' \quad (11)$$

The first term in this equation can be converted to a surface integral along the boundary of the domain using the divergence theorem. The second term, which contains $\nabla_{\mathbf{r}'} W(|\mathbf{r} - \mathbf{r}'|)$ - the gradient of the kernel evaluated at position $\mathbf{r}'$ - can be rewritten, noting that the symmetry of the kernel function means that the gradient of the kernel is anti-symmetric. Therefore, flipping the evaluation of the gradient changes its sign: $\nabla_{\mathbf{r}'} W(|\mathbf{r} - \mathbf{r}'|) = -\nabla_{\mathbf{r}} W(|\mathbf{r}' - \mathbf{r}|)$. The equation can then be rewritten

$$\nabla A(\mathbf{r}) = \int_{\partial\Omega} A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|) \mathbf{n}(\mathbf{r}') dS + \int_{\Omega} A(\mathbf{r}') \nabla_{\mathbf{r}} W(|\mathbf{r} - \mathbf{r}'|) d\mathbf{r}' \quad (12)$$

The first term is now a surface integral over $\partial\Omega$ - the *boundary* of the domain of the kernel, and $\mathbf{n}(\mathbf{r}')$ is the outward pointing normal of the surface element dS . If the kernel has a finite support as discussed, then this means $W(|\mathbf{r} - \mathbf{r}'|)$ is identically zero along the boundary of the domain (as long as the domain is not truncated) and so the first term in Eq. (12) is often set to 0. Finally, applying the discrete Riemann sum approximation to the second term, we find the most basic gradient estimate in SPH:

$$\nabla A_i = \sum_j V_j A_j \nabla_i W_{ij} \quad (13)$$

where $W_{ij} := W(|\mathbf{r}_{ij}|)$, $\mathbf{r}_{ij} := \mathbf{r}_i - \mathbf{r}_j$ as a shorthand. This form derived can be equivalently applied to a scalar field or vector field. The SPH gradient estimate manages to transfer the gradient operator from the function onto the kernel, which is advantageous as the gradient of the kernel is known *a priori* exactly. It is also noted that the properties of the kernel function mean that $\nabla_i W_{ij} \equiv \nabla W(|\mathbf{r}_{ij}|) \mathbf{e}_{ij}$ where ∇W is the derivative of W considered as a function of the scalar variable $|\mathbf{r}_{ij}|$, and $\mathbf{e}_{ij} := \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}$. This form is often preferential when implementing the equation in a numerical code.

The gradient form of Eq. (13) can be extended to the divergence following the same methodology, leading to

$$\nabla \cdot \mathbf{A}_i = \sum_j V_j \mathbf{A}_j \cdot \nabla_i W_{ij} \quad (14)$$

These two Eqs (13) (14), along with the SPH equation to evaluate a variable (2), then form the basis of more complete and complex formulations of SPH estimates.

Indeed, the gradient approximation presented in Eq. (13) is not the most accurate nor very commonly used. Different forms can be found by rewriting the gradient of a function as

$$\nabla A = \frac{1}{\phi^k} (\nabla (\phi^k A) - A \nabla \phi^k) , \quad (15)$$

$$\nabla A = \phi^k \nabla \left(\frac{A}{\phi^k} \right) + \frac{A}{\phi_k} \nabla \phi^k \quad (16)$$

where ϕ is any differentiable scalar quantity defined on the particles, typically taken as the density $\phi = \rho$, and k is any real number. Equation (15) is used to produce symmetric formulations of gradient related quantities, and Eq. (16) is used to produce anti-symmetric formulations. Eqs (15) and (16) can be extended to discrete SPH estimates by appropriately applying Eq. (13) or (14), and noting that any variables inside the ∇ operator should be taken over a sum of neighbouring particle indices j whilst any variable outside the operators should be taken as the current particle index i . Divergence, curl, and other gradient related quantities can also be found following an equivalent method. Divergences, for example, can be modelled as

$$\nabla \cdot \mathbf{B} = \frac{1}{\phi^k} (\nabla \cdot (\phi^k \mathbf{B}) - \mathbf{B} \cdot \nabla \phi^k) , \quad (17)$$

$$\nabla \cdot \mathbf{B} = \phi^k \nabla \cdot \left(\frac{\mathbf{B}}{\phi^k} \right) + \frac{\mathbf{B}}{\phi_k} \cdot \nabla \phi^k \quad (18)$$

These formulations allow us to produce gradient, divergence, and other derivative estimates with desirable properties. Other gradient operators which minimise the linear error based on a basic Taylor series analysis such as in Sec. 2.3 can also be found. These can be extended to higher order in order to increase the order of accuracy of the SPH estimate. Interestingly however, these higher-order estimates will not always produce the most accurate results. This is because minimising linear errors does not address the *non-linear* properties of the system, such as symmetry and conservation. Eqs (15) and (16) however, do have the benefit of producing beneficial symmetry and conservation properties despite their linear errors. In fact, the linear error terms in these estimates have a regularising effect which tends to re-order particles to a regular distribution and minimise the errors discussed in Sec. 2.3 [130, 156]. In this way, these “low-order” estimates manage to maintain a high accuracy.

It also possible to derive SPH formulations using Lagrangian and Hamiltonian dynamics, which explicitly conserve momentum and energy. In this case, a system of equations is

produced which is identical to those produced when using Eqs (15) and (16), further justifying this formulation [130].

A further guide when choosing gradient and divergence operators (the two more prevalent operators in a fluid dynamics model) is that they should be skew-adjoint: $\langle \nabla A, \underline{B} \rangle = -\langle A, \nabla \cdot \underline{B} \rangle$, where A is any scalar field and B is any vector field and $\langle \rangle$ is the L^2 scalar product. This is because the energy of the system is exactly conserved in time if the gradient and divergence operators are skew-adjoint. A full discussion of this can be found in [104]. In general, this can be done by taking an anti-symmetric formulation of the gradient, and a symmetric formulation of the divergence, and selecting either $k = 0$ or $k = 1$ consistently. This then produces the two most commonly employed gradient and divergence schemes in SPH models:

$$\begin{aligned}\nabla A_i &= \sum_j V_j (A_i + A_j) \nabla_i W_{ij} \\ \nabla \cdot \mathbf{B}_i &= - \sum_j V_j (\mathbf{B}_i - \mathbf{B}_j) \cdot \nabla_i W_{ij}\end{aligned}\tag{19}$$

and

$$\begin{aligned}\nabla A_i &= \rho_i \sum_j \rho_j V_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla_i W_{ij} \\ \nabla \cdot \mathbf{B}_i &= - \frac{1}{\rho_i} \sum_j \rho_j V_j (\mathbf{B}_i - \mathbf{B}_j) \cdot \nabla_i W_{ij}\end{aligned}\tag{20}$$

For the work carried out during this thesis, it was found that in practice there was not much difference between these two implementations. However, some literature has found that the system (19) can produce fewer fluctuations in the pressure field when simulating interfacial flows [30]. This is less relevant for the work in this thesis as at no point were any interfacial (fluid and air) flows simulated. It should be noted that the system (20) does produce governing equations of fluid dynamics equivalent to a Lagrangian/Hamiltonian derivation, thus clarifying the strong conservation properties that have been so far discussed.

2.5 Fluid Dynamics equations

The equations of incompressible fluid dynamics - the momentum and continuity equation - that will govern our system are:

$$\begin{aligned}\frac{d\rho}{dt} &= -\rho \nabla \cdot \mathbf{u} \\ \frac{d\mathbf{u}}{dt} &= -\frac{\nabla P}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{g} + \frac{\mathbf{f}}{m}\end{aligned}\tag{21}$$

Here P is the pressure, $\mathbf{u}$ is the velocity, ν is the kinematic viscosity of the fluid, $\mathbf{g}$ is gravity, and $\mathbf{f}$ is any other body-force acting on the fluid, such as the coupling force with solids.

As previously mentioned, the forms of gradient and divergence typically used (Eqs (19) (20)) entail an anti-symmetric gradient and symmetric divergence for conservation properties. This can be seen conceptually in looking at the governing equations (21). The gradient as it appears in ∇P is a force term, and so should ideally be anti-symmetric such that Newton's Third Law is satisfied (force on particle i from particle j is equal and opposite to the force on particle j from particle i) and momentum is conserved. It may seem strange that the anti-symmetric formulation of ∇P is preferred when considering that it does not produce 0 force for a constant pressure field. However, looking at the total system involving all particles, all forces would be equally matched and so the net force would still remain 0. This is analogous to what happens in real pressurised fluid systems, where there is always a constant pressure force at every point, it is just that it is balanced in all directions which leads to a motionless system.

A symmetric formulation of $\nabla \cdot \mathbf{u}$ has the benefit of ensuring that a constant velocity field does not produce any divergence, and in fact further analysis shows that it ensures energy conservation [32].

The Laplacian ∇^2 is a second-order derivative and here represents the effect of viscosity. It is possible to extend the method applied to the first-order derivative and just use the second derivative of the kernel:

$$\nabla^2 \mathbf{C}_i = \sum_j V_j \mathbf{C}_j \nabla_i^2 W_{ij}. \quad (22)$$

However, this form is very sensitive to particle disorder. The second derivative of the kernel has more unfavourable properties than the first derivative - it changes from negative to positive with distance, and is discontinuous in some cases. This means the domain must be very well sampled to avoid undesirable effects [114]. Noting that $\nabla^2 \mathbf{A} = \nabla \cdot \nabla \mathbf{A}$, the definitions in Eq. (19) can be applied in succession to create a formula for the Laplacian which involves a two-fold sum. In practice, this is not often used due to the computational expense.

Instead, one of the more commonly employed formulations, which is used in this work, is the so-called ‘‘Morris’’ operator [119] which combines an SPH divergence approximation with a finite difference approximation for the inner gradient. First an anti-symmetric first order divergence following the formulation in Eq. (18) with $k = 0$ is applied to the gradient. The Laplacian can thus be written

$$\nabla^2 \mathbf{A}_i = \sum_j V_j (\nabla \mathbf{A}_i + \nabla \mathbf{A}_j) \cdot \nabla_i W_{ij} \mathbf{e}_{ij} \quad (23)$$

Here, instead of approximating $\nabla \mathbf{A}_i$ with a discrete sum approximation, linearisation of the function $\mathbf{A}$ can be used: $\nabla \mathbf{A}_i \cdot \mathbf{e}_{ij} \approx \frac{\mathbf{A}_i - \mathbf{A}_j}{|\mathbf{r}_i - \mathbf{r}_j|}$, which is found from a Taylor series expansion of $\mathbf{A}_j$ about $\mathbf{r}_i$. Further it can be noted that

$$\nabla \mathbf{A}_j \cdot \mathbf{e}_{ij} = -\nabla \mathbf{A}_j \cdot \mathbf{e}_{ji} = \nabla \mathbf{A}_i \cdot \mathbf{e}_{ij} \quad (24)$$

Applying this, along with the finite difference approximation of $\nabla \mathbf{A}_i$ and $\nabla \mathbf{A}_j$ in Eq. (23), leads to the aforementioned Morris operator

$$\nabla^2 \mathbf{A}_i = 2 \sum_j V_j \frac{\mathbf{A}_{ij}}{|\mathbf{r}_{ij}|} \nabla W_{ij}. \quad (25)$$

As with first-order derivatives, the Laplacian is equivalently open to different formulations, which offer different consistency and accuracy advantages [114, 171]. Of note is that the Morris Laplacian in Eq. (25) suffers inconsistencies near the free surface [139], though the correction methods available involves inverting matrices. These are relatively expensive whilst not providing much increase in accuracy for the problems of interest. For this reason they are not employed in this work.

With a combination of the given formulations, the fluid dynamics equations (21) can be approximated as:

$$\frac{d\rho_i}{dt} = \sum_j \rho_j V_j \mathbf{u}_{ij} \cdot \nabla W_{ij} \mathbf{e}_{ij} \quad (26)$$

$$\begin{aligned} \frac{d\mathbf{u}_i}{dt} = & - \sum_j \rho_j V_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla W_{ij} \mathbf{e}_{ij} \\ & + 2\nu \sum_j V_j \frac{\mathbf{u}_{ij}}{|\mathbf{r}_{ij}|} \nabla W_{ij} \\ & + \mathbf{g} + \frac{\mathbf{f}_i}{m_i} \end{aligned} \quad (27)$$

It should be noted that the above continuity and momentum equations are not closed - the pressure term in the momentum equation (27) is unknown. This means we require at least one more equation for the pressure. For the typical incompressible flow, the partial

differential equation for pressure - the Poisson equation - is used. For a compressible flow, a state equation is used which relates the pressure and density of the fluid. The pressure treatments will be discussed in Sec. 2.6.1.

Eqs (26) and (27) are just one way of representing the governing equations of SPH, as will be seen if the reference works are read. The variations in representation largely come down to the various formulations of the derivative operators which have been discussed. There are also numerical techniques and additions that can be used to adjust formulations to improve accuracy or stability of the method. Some of these further alternate formulations and methods will be discussed in the following section.

2.6 Further techniques

There are a wide range of different techniques and further formulations that can be applied to the SPH method, many of which are still being currently developed. As with any CFD code, some of these additional techniques encompass numerical dissipation necessary to help stabilise the method and improve accuracy. It is often the case that different problem types benefit from different adjustments to the method, and the ideal formulation for any specific problem is still an open question. Here we will discuss some of the most integral and widely-used techniques, as well as the ones deemed most relevant to the coastal dynamics problems of interest.

Handling of solid boundaries is an integral technique within SPH, and will be addressed in a separate section 3. Mobile boundaries, which involve coupling with a solid modelling technique, will also be addressed in a later section 6.

2.6.1 Density and Pressure

Density summation

So far we have only presented the time derivative of density deriving from the continuity equation Eq. (26). Using the standard SPH approximation for a variable (Eq. (2)) it is also possible to write the density as

$$\rho_i = \sum_j \rho_j V_j W_{ij} \quad (28)$$

This formulation of density is effectively a rewriting of the conservation of mass. It can also be seen as equivalent to the continuity equation by taking the time derivative, which reproduces Eq. (26). This means there are generally two methods for updating the density in an SPH simulation - directly summing over local particles - position dependent -

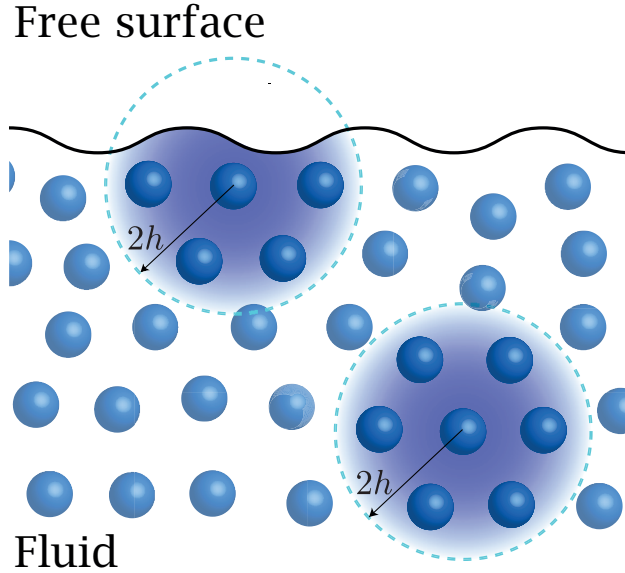


Figure 3: Visualisation of the SPH interpolation near a free surface.³ The void of particles on one side of the free surface leads to an underestimation of values calculated by a direct sum.

or integrating in time based on the local velocity - time dependent. The position dependent formulation would theoretically be preferable, as it conserves mass and has no potential time integration errors.

An important caveat is that Eq. (28) performs very poorly in free surface problems. At a free surface, the kernel support is incomplete due to the void of particles at the free surface, and so the stipulations that ensured second-order accuracy in Sec. 2.3 are no longer fulfilled. A visualisation of the SPH interpolation issue near a free surface can be seen in Fig. 3. Any value near a free surface calculated based on a direct sum will be poorly sampled and produce incorrect results. Even in the case where the free surface is modelled as air particles, the smoothing nature of SPH will smooth the density values near the free surface and not accurately model the density step. For this reason, in free surface models a time-dependent density from the continuity equation is almost always used.

There are a few available ways to combat this issue though. Renormalising the kernel can ensure that zeroth and first order accuracy are restored no matter where in the domain, as well as reframing the density to not be dependent on surrounding masses but only the mass at the given point $\rho_i = m_i \sum_j W_{ij}$ [72].

The so-called “Shepard Filter” [17] simply renormalises the kernel function such that it always sums to 1 over its domain.

³Fluid particles in an SPH simulation are actually just point particles, and their “radius” is better understood as a radius of influence given by the support radius $2h$. Particles are represented with a finite volume in schematics for visualisation purposes.

$$\rho_i = \frac{1}{\sum_j V_j W_{ij}} \sum_j \rho_j V_j W_{ij}. \quad (29)$$

This is equivalent to dividing by the discrete form of the first term on the RHS of Eq. (5), and restores zeroth-order accuracy. This filter can either be applied consistently, if a position-based density is used; or, if a time-dependent density is used, then the filter can still be applied periodically every N time steps, where N is on the order of 30. This acts as a numerical diffusion, helping to smooth the density field every N time steps which can reduce errors accumulated during time integration.

It is also possible to ensure 1st order accuracy, but this requires the matrix inversion of the discrete form of the second term on the RHS of Eq. (5) [41]. However, this is markedly more expensive than the zeroth-order correction, requiring an extra matrix to be stored at each particle and a matrix inversion operation.

Compressibility

As briefly mentioned in Sec. 2.5, there are in general two ways to close the equations (26) and (27) and determine the pressure in fluid dynamics system - compressibly or incompressibly.

In most SPH implementations, a weakly-compressible SPH (WCPSH) method is employed. This involves using an equation of state which links the pressure to the density. In this way, the fluid is technically treated as compressible, though with a such a low range for compressibility that it behaves in a nearly incompressible manner. The equation of state most often used for fluids like water is the Tait, or Cole, equation,

$$P(\rho) = \frac{\rho_0 c_0^2}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right), \quad (30)$$

where γ is a stiffness parameter, and ρ_0 is the reference density - a predefined constant depending on the fluid and not the same as ρ which represents a specific fluid particle's variable density. Based on physically realistic parameters for water, $\gamma = 7$ and $\rho_0 = 1000 \text{ kgm}^{-3}$. The speed of sound, c_0 , is defined at the reference density, and in theory could be the real physical speed of sound associated with the fluid. For example, for water, $c_0 = 1480 \text{ ms}^{-1}$. In the case of SPH however, the speed of sound is not taken at the real value, but rather viewed as a numerical parameter. A physically realistic value could be chosen, but this would lead to several issues. Very large speed of sounds can lead to a locking phenomena in steady-state SPH solutions [146], where a lack of compressibility means that small oscillations inherent to SPH induce similar motion in nearby particles, directly transferring the oscillations throughout the system and thus becoming magnified. Generally, however, for transient, energetic simulations, an increased speed of sound can lead to more accurate results. The most notable

issue for large speeds of sound, however, are the reduction in timestep they necessitate, as such leading to prohibitively slow simulations. This will be seen in Sec. 2.7.1.

Therefore, a numerical speed of sound is used, which is much lower, but still large enough to limit compressibility of the system. Typically, this is set as 10 times the maximum fluid velocity U_{max} , corresponding to a Mach number $M \sim 0.1$. This limits density fluctuations to $\sim 1\%$ [161] (as density fluctuations scale as M^2), and thus retains a nearly incompressible behaviour [114]. It is not always possible to accurately know the maximum fluid velocity *a priori*, but this can either be found through test simulations or through physical considerations of the system. In the case of gravity-driven flows, the characteristic velocity is given by $\sqrt{gH}$ [135, 161], where H is the maximum height of fluid in the system - a characteristic vertical dimension of the flow. As such it is preferable to set the speed of sound of the system as

$$c_0 = 10 \max \left(U_{max}, \sqrt{gH} \right) \quad (31)$$

Incompressible SPH (ISPH) solves the Poisson pressure equation with a projection method, matching the predicted velocity to the divergence-free velocity field $\nabla \cdot \mathbf{u} = 0$. This has to be solved implicitly, and is more computationally expensive to solve than the standard SPH equations, but does ensure a more accurate pressure field which matches the expected incompressible dynamics of real water systems.

Weakly compressible dynamics are perhaps an intuitive approach for SPH. The free motion of the particles lends itself nicely to encoding a weakly compressible system, where particles can coalesce and spread out producing high and low density areas respectively, and as a result high and low pressure areas. A collision of fluid against a boundary, for example, would cause particles to flow more closely together and thus lead to an increase in pressure, as would be expected. Pressure gradients are created which counteract the gathering/spreading of particles and act to regularise their relative positions. This mechanism ensures minimal density variations are generally maintained, but they can be exceeded in high velocity impacts.

WCSPH can also explicitly solve free surfaces, as the density drops to 0 at a free surface, and the free-surface boundary condition of $P = 0$ is automatically satisfied. Along with its relative ease of implementation and completely explicit solutions, this has made WCSPH a more popular method than ISPH.

However, WCSPH is not without its disadvantages. The numerical speed of sound confers a very limited time step for the system, as will be seen in Sec. 2.7.1. This means it can be very slow despite the ease of calculations. More importantly, the stiffness of the equation of state (30) means that very small density fluctuations can lead to very large pressure

fluctuations, and so small errors in the density lead to very large errors in the pressure. This issue is especially pertinent when considering high velocity fluid impacts against solid objects or the presence of solids affecting local fluid density, such as is common in coastal dynamics. The large density or pressure fluctuations associated with these create at best a very noisy pressure field, and at worst instabilities in the system.

ISPH is just one of the methods that was developed in response to this. ISPH offers advantages in more accurate densities and pressures, and occasionally offers improved efficiency [73]. However, ISPH is more complicated to implement, and the improved accuracies are often only evident in certain systems. When applying certain smoothing procedures and filtering techniques to WCSPH it is found that it performs just as well if not better than ISPH [73]. It has therefore also been a route of development to directly improve the WCSPH method by reducing density errors and oscillations.

Improving pressure fields

This can be done for example using the Shepard filter as discussed earlier in this section, which acts as a numerical diffusion to smooth the density field and reduce large fluctuations. The viscosity term in the governing equations is a physical diffusion term, which also helps to smooth the velocity field and thus help reduce noise which manifests as velocity fluctuations. This is often replaced through use of an *artificial viscosity*, which will be discussed in Sec. 2.6.2. Viscosity terms are usually not enough to completely stabilise the system, and so it has now become standard practice to apply an artificial diffusion term to the density computation. Numerical diffusive terms in SPH are well-discussed in literature by Antuono et al. [11, 9].

It can be considered that the equation of state used in WCSPH (30) is isothermal, and so the evolution of energy is decoupled from the mass and momentum equations unlike in physical systems. An appropriate diffusion term added to the density equation reduces short-scale fluctuations and thus avoids the non-physical energy flux to high frequency modes. More simply, a diffusion term helps to smooth the density field. An artificial diffusion term is the preferred method for smoothing the density field since the SPH diffusion operator is conservative and does not otherwise affect the actual solution [11, 9].

One of the more well-developed and commonly used density diffusion schemes is δ -SPH [99], which involves a density diffusion term of the form

$$\delta h c_0 \sum_j \psi_{ij} \cdot \nabla_i W_{ij} V_j \quad (32)$$

where

$$\psi_{ij} = 2 \frac{(\rho_i - \rho_j)}{|\mathbf{r}_{ij}|} \mathbf{e}_{ij} \quad (33)$$

and δ is a tunable parameter to control the strength of the diffusion effect, though has a limited stability range and is usually fixed as 0.1 [11]. Considering Eq. (25) it can be seen that this Eq. (32) is effectively a form of $\nabla^2 \rho$, hence the understanding as a density diffusion term. The factor of h ensures that diffusion reduces to zero as resolution increases, and in fact most numerical additions to the SPH scheme are designed to be $\mathcal{O}(h)$ for this reason.

The δ -SPH scheme is often improved by restoring first-order accuracy to the diffusion term through the solution of a renormalisation matrix [151], similarly to the density summation. This can be a valuable process to satisfy convergence throughout the whole fluid domain, since it has been shown that the inclusion of a density diffusion term leads to spurious forces at a free surface if not appropriately renormalised, and will not be able to reproduce the hydrostatic solution [11, 9]. However, this is not done in this work as the process is fairly computational expensive, requiring a matrix to be summed and inverted at each particle before ψ_{ij} can be calculated.

Instead, Eq. (33) can be more simply improved by rewriting the total density difference $(\rho_i - \rho_j)$ as the dynamic density difference $(\rho_i^D - \rho_j^D)$. This comes from understanding that the total density of a fluid particle ρ can be broken down into a hydrostatic component ρ^H depending on the hydrostatic pressure P^H , and a dynamic component ρ^D depending on the fluid flow (dynamic pressure).

This acts to redefine

$$\psi_{ij} = 2 \frac{(\rho_i^D - \rho_j^D)}{|\mathbf{r}_{ij}|} \mathbf{e}_{ij} \quad (34)$$

The dynamic density difference can be calculated as

$$\begin{aligned} \rho_i^D - \rho_j^D &= \rho_{ij} - \rho_{ij}^H \\ \rho_{ij}^H &= \rho_0 \left(\frac{\gamma P_{ij}^H}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}} \\ P_{ij}^H &= \rho_0 (\mathbf{g} \cdot \mathbf{r}_{ij}) \end{aligned} \quad (35)$$

where the second line in Eq. (35) is just an inverted form of the equation of state (30). By diffusing only the dynamic density, a hydrostatic solution can be maintained and behaviour near a free surface is improved whilst still filtering the majority of high frequency density oscillations. This is the same scheme as has been adopted as some other SPH implementations [58].

Using this model for density diffusion, the continuity equation (26) instead becomes

$$\frac{d\rho_i}{dt} = \sum_j V_j (\rho_j \mathbf{u}_{ij} + \delta h c_0 \psi_{ij}) \cdot \nabla W_{ij} \mathbf{e}_{ij} \quad (36)$$

The improvements gained from such a small adjustment to the density evolution are very impressive and have been well documented in literature [111, 11, 99].

Pressure initialisation

It is often preferable in a simulation to initialise the appropriate initial conditions of the fluid. If a flow is not initialised properly then this is akin to a large force discontinuity at the start of the simulation, which can cause large unphysical pressure spikes. Similarly, if the fluid is not placed in the appropriate location where the initial conditions are satisfied, then it may have to “fall” into the equilibrium position, in the process generating forces and spurious energies.

This is especially the case for SPH, where the weak compressibility means the SPH particles will first compress under the initial forcings, then generate a pressure field, then decompress, and oscillate between these states until the steady state initial conditions are reached. This is essentially the fluid particles iteratively solving their configuration until they reach the stable lowest energy state.

These issues are of course only particularly relevant for systems which require some kind of steady-state initial conditions, whether on the pressure or velocity fields.

The typical example of this is a stationary box of fluid under hydrostatic equilibrium. As the particles attempt to settle under gravity into hydrostatic equilibrium, they produce an oscillating pressure field. This leads to a very noisy pressure field that can take a long time to settle. This behaviour is greatly reduced by initialising the fluid to a hydrostatic pressure gradient and appropriately placing the particles. This is equivalent to improving the initial guess to an iterative solver, which allows the system to converge to the stable solution more quickly. An example can be seen in Fig. 4, which compares the behaviour of fluid in a box with hydrostatic initialisation versus the default zero pressure field initialisation.

In a stationary box of hydrostatic fluid, the expected pressure at any point can be calculated from $P_i^* = \rho_0 g h_i$, where h_i is the height of the fluid at the given point. The relative error in pressure calculation can then be given by

$$P_{rel} = \frac{P_i - P_i^*}{P_i} \quad (37)$$

where the error is normalised with respect to the calculated pressure P_i rather than the expected pressure P_i^* in order to avoid divisions by zero. In Fig. 4, the plot on the left

shows the evolution of relative error in pressure. It can be seen that whilst both methods have some noise as they converge to their steady state, the uninitialised fluid exhibits much larger errors in pressure and takes longer to converge. The effects of this noisy pressure field can also be seen in the plot on the right in Fig. 4, where the kinetic energy of the system spikes and oscillates in the early timesteps. In a stationary box of fluid, we expect the total kinetic energy to be close to zero, as none of the particles should be moving. Again, whilst both methods display a spike in energy at the beginning of the simulation as they settle into their stable states, the uninitialised fluid exhibits much larger kinetic energies, and remains more energetic than the initialised fluid throughout. The implication of this is that any hydrostatic solution should be initialised appropriately to avoid the initial noisy pressure field and spikes in energy. An alternative, and perhaps preferred in some instances, is that the fluid should be allowed to settle until the initial oscillations have died out before the following required dynamics in the simulation are added. This is good practice in general for SPH, since any simulation which is started in a steady-state will inevitably involve some resettling of the particles and consequent noise in the pressure field. This can affect other dynamics in the simulation if the fluid is not first allowed to settle.

This resettling behaviour at the initialisation of an SPH simulation is also affected by the numerical speed of sound. A larger speed of sound will allow faster convergence to the steady-state solution. Similarly, smoothing techniques, such as those mentioned in this work, can also allow for a reduction in oscillation noise at the onset of a simulation and improved convergence times.

A fluid can be simply initialised to a hydrostatic pressure gradient by assigning an initial hydrostatic pressure $P_{i,t=0} = \rho_0 (\mathbf{g} \cdot \mathbf{x}_i)$ to each particle, where $\mathbf{x}_i$ is the particle's position relative to an initial free surface reference height. The positions of the particles should also be suitably adjusted. As we are increasing the pressure of some particles, this increases their density based on the equation of state (30), and thus if they are to maintain a constant mass m_0 , their initial volume $V_{i,t=0} = dx_{i,t=0} dy_{i,t=0} dz_{i,t=0}$ should be suitably adjusted. This is not to say that V_j used throughout the SPH summations is changed (indeed this is a constant for all particles), but just that the particles' initial spacing should appropriately reflect their increased initial density under the weakly compressible assumption. Since density increases are generated by gathering of particles closer than their initial default spacing, we expect that particles under a hydrostatic equilibrium with increased pressure will be slightly compressed and more closely spaced the deeper they are. The new initial particle spacings for a hydrostatic equilibrium can be calculated following

$$\begin{aligned}
P_{i,t=0} &= \rho_0 (\mathbf{g} \cdot \mathbf{x}_i) \\
\rho_{i,t=0} &= \rho_0 \left(\frac{\gamma P_{i,t=0}}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}} \\
m_i &= m_0 = \rho_0 V_0 \\
m_i &= \rho_{i,t=0} V_{i,t=0}
\end{aligned} \tag{38}$$

The open-source code SPHysics [63] also offers this same density and mass initialisation.

We can also define $V_0 = dx \cdot dy \cdot dz$ and $V_{i,t=0} = dx_{i,t=0} \cdot dy_{i,t=0} \cdot dz_{i,t=0}$. Then, combining this set of Eqs (38) we find

$$\begin{aligned}
\rho_0 \left(\frac{\gamma P_{i,t=0}}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}} V_{i,t=0} &= \rho_0 V_0 \\
V_{i,t=0} &= \frac{V_0}{\left(\frac{\gamma P_{i,t=0}}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}}} \\
dx_{i,t=0} \cdot dy_{i,t=0} \cdot dz_{i,t=0} &= dx \cdot dy \cdot \frac{dz}{\left(\frac{\gamma P_{i,t=0}}{\rho_0 c_0^2} + 1 \right)^{\frac{1}{\gamma}}}
\end{aligned} \tag{39}$$

Thus, to ensure a constant mass and appropriate volume for a compressed particle, the particle's initial spacing should be reduced by a factor of the relative density increase. This scaling is logically applied in the same direction as the compressive force, in this case to the dz spacing, as in Eqs (39). This spacing serves as a more accurate initial condition for the particle spacing under hydrostatic equilibrium.

2.6.2 Viscosity

Whilst the viscosity term was already formulated in Sec. 2.5, it is often the case that inviscid flow is considered and viscous stresses are disregarded. Though, for accuracy or stability, a viscosity term may still be used for its dissipative effects as a diffusion term in the momentum equation. It is generally found that the accuracy and stability of SPH is much more greatly impacted by diffusion terms in the continuity equation as was presented in Sec. 2.6.1.

There are many formulations of the viscous stresses that may be used [171]. One such example, originally developed for stability purposes, is the commonly used artificial viscosity [113]. Using an artificial viscosity formulation, the momentum equation (27) becomes

$$\frac{d\mathbf{u}_i}{dt} = - \sum_j \rho_j V_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} + \Pi_{ij} \right) \nabla W_{ij} \mathbf{e}_{ij} + \mathbf{g} \tag{40}$$

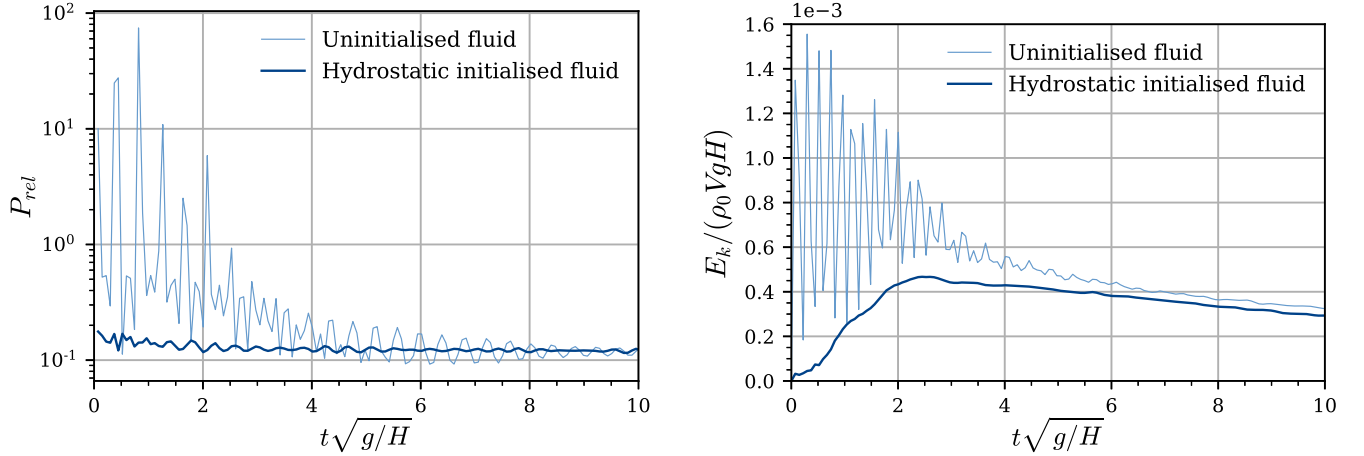


Figure 4: Comparison of the behaviour of a stationary box of fluid with no pressure initialisation versus hydrostatic pressure initialisation. The plot on the left shows the evolution of the average relative error of pressure in the system, whilst the plot on the right shows the evolution of the total kinetic energy in the system. Time is represented in non-dimensional form.

where the viscosity term is given by

$$\Pi_{ij} = \nu \left(\frac{\mathbf{u}_{ij}}{r_{ij}} \mathbf{r}_{ij} r_{ij}^2 + \epsilon h^2 \right) \quad (41)$$

where $\nu = \alpha h \bar{c}_{ij} / \bar{\rho}_{ij}$, and the overbar represents the arithmetic mean of the values at i and j . The local speed of sound is $c_i = \sqrt{P_i / \rho_i}$. ϵ is introduced to avoid a singularity when $r_{ij} \approx 0$, and α is a constant tuned to the amount of dissipation required. This implementation is commonly found in open-source SPH codes, primarily for its simplicity.

Following the work in Sec. 2.5, we can apply the Laplacian (25) to the velocity vector in order to find a viscosity formulation as in Eq. (27). This is a reasonable choice to model laminar viscous stresses for low to moderate Reynolds numbers, as it is a direct SPH approximation to $\nu \nabla^2 \mathbf{u}$ as is found in the governing equations (21). Similar to the artificial viscosity formulation, this also has the effect of providing small stabilisation and dissipative properties to the scheme.

Since the laminar viscous stress formulation of viscosity follows simply from the governing equations, this is the choice implemented in the current code. Though it has a physical analogue, this is not to say that the kinematic viscosity ν need always be set to the physically accurate value. It remains an approximation to the modelling of viscous effects, and while the physically accurate choice of ν is often a good starting point, there is always scope for it to be adjusted depending on the dynamics of the flow or in order to match validation tests. It is noted that the choice of viscosity model was found to have almost no noticeable impact

on the overall dynamics of the problems of interest.

There are alternative viscosity treatments available in the literature, such as the Sub-Particle-Scale approach [37] which attempts to model the effects of turbulence below sub-particle scale, and also implementations of the Large Eddy Simulation model in SPH [64]. In general, the problems of interest for this work were found to not be largely affected by the viscosity modelling, and hence a further discussion of viscosity or turbulence modelling is not presented, though perhaps considered as future work.

2.6.3 Particle Shifting

The accuracy of the SPH interpolation is largely dependent on the regularity of the particles [31]. The error terms discussed in Sec. 2.3 only hold true, and are in fact minimised, for regular symmetric arrays of particles. Many studies have been carried out aiming to quantify the effects of particle disorder on the various SPH approximations [132, 7, 10]. The self-regularising effect of SPH tends to maintain regularity in the particle distribution, but this is not always the case - particularly in high velocity flows involving impacts, such as those we are interested in. To this end, it is often desirable to introduce techniques which supply additional controlled accelerations or movement to maintain particle regularity.

In highly energetic flows involving boundary discontinuities, it is often the case that unphysical voids can be formed due to particle clustering effects present when particles are highly disordered [110]. This can be remedied through the use of a particle shifting technique, which aims to very slightly shift particles based on a diffusion-like process in a way that does not greatly affect the conservative properties of the scheme, whilst still ensuring a more regular spacing of particles for improved stability and accuracy [90, 145, 152]. The diffusion like particle shifting is written as

$$\delta \mathbf{r}_i = -\mathcal{D}_i \nabla C_i \quad (42)$$

where

$$C_i = \sum_j V_j W_{ij} \quad (43)$$

is the local particle concentration and its gradient can be found by a typical SPH approximation such as $\nabla C_i = \sum_j V_j \nabla W_{ij} \mathbf{e}_{ij}$. This term points in the direction of maximum anisotropy [31], and as such ensures that the particle shifting is always directed from regions with the highest concentration of particles to regions with the lowest concentration of particles, satisfying the desired regularising effect.

The strength of the process is controlled by $\mathcal{D}_i$, which is a numerical diffusion coefficient. Whilst there are some different forms for this, the form used in this work follow that of Skillen et al. [145], where $\mathcal{D}_i = Ah \|\mathbf{u}_i\| \Delta t$, and $A = 2$. The value of A should be set fixed and not tuned, as the value of 2 arises due to the Courant condition and is thus an intrinsic stabiliser on the particle shifting.

Due to large concentration gradients at the free surface, applying a particle shifting of the form in Eq. (42) would produce spurious results, with particles at the free surface being diffused into the void around them. To remedy this effect, Lind first proposed a correction which allowed control of the local normal and tangential particle shifts near the free surface, and set the normal direction particle shift at the free surface to 0 for high speed flows such as dam breaks [90]. It is also possible to just reduce or remove shifting for particles on or near the free surface.

In order for any free surface adjustment to be made it is of course necessary to identify free-surface particles. Solely using the concentration or density of particles is not sufficient, since the weakly compressible framework means that these values often oscillate around their relative values during the slight compression and expansion in high-speed flows. Some free surface detection algorithms are quite advanced, and notably require implicit calculations such as the inversion of a matrix and calculating its eigenvalues [152]. Instead, in this work, a more simple but still very effective free surface detection method based on the work by Skillen et al. [145] is used, which reads

$$\begin{aligned} \nabla \cdot \mathbf{r}_i &\leq n - 0.5 \\ \nabla \cdot \mathbf{r}_i &= - \sum_j V_j \mathbf{r}_{ij} \cdot \nabla W_{ij} \mathbf{e}_{ij} \end{aligned} \tag{44}$$

where n is the dimension of the problem. This states that any particle whose position divergence is below a certain threshold can be identified as belonging to the free surface. Indeed, for a problem in n -dimensions, a particle in the bulk of the fluid will have a position divergence equal to n , and so a criterion of $\leq n - 0.5$ is an effective and efficient method of identifying the free surface whilst remaining in a straightforward fully-explicit implementation.

Turning off particle-shifting for free-surface particles and otherwise fully evaluating the term presents a discontinuity and can lead to clustering of particles near the free surface. In order to avoid this, many of the implementations have incorporated some form of scaling in the particle-shifting term based on a particle's proximity to the free surface. Skillen et al. for example used a sinusoidal-shaped scaling function based on $\nabla \cdot \mathbf{r}_i$. In this work, different scaling functions were found not to have a large impact on the behaviour of the

free-surface shifting, and $\nabla \cdot \mathbf{r}_i$ was not found to vary in a consistent enough manner from free surface to fluid bulk. For this reason, and for simplicity and computational speed, a simple linear scaling from 0 to 1 is applied to $\delta \mathbf{r}_i$ for particles that interact with a free-surface particle based on their distance to the free surface. This does require first calculating which particles belong to the free surface (which is usually done in an initial loop over the particles whilst calculating normalisation terms) and then calculating the minimum distance to a free-surface particle $d_{\mathcal{FS},i}$ and the free-surface scaling ϕ_i for any particles within $2h$ of a free-surface particle. This can be done as a relatively cheap additional calculation during the main loop for calculating particle interactions, since the particle shift need only be applied on particle update at the end of a time step.

The final scheme then reads

$$\delta \mathbf{r}_i = -\phi_i \mathcal{D} \nabla C_i \quad (45)$$

with

$$\phi_i = \begin{cases} 0 & \text{if } \nabla \cdot \mathbf{r} \leq n - 0.5 \\ \frac{d_{\mathcal{FS},i}}{2h} & \text{if } d_{\mathcal{FS},i} \leq 2h \\ 1 & \text{otherwise} \end{cases} \quad (46)$$

Since the shift of particle positions $\delta \mathbf{r}_i$ produces only very small perturbations relative to the particle spacing, it is generally not necessary to equivalently shift the velocities and densities based on this change [91]. In this way the particle shifting can be considered as a non-Lagrangian transport velocity.

It should also be noted that whilst initially designed to counteract unphysical void formations and other particle clustering effects, particle shifting can have general benefits even when these effects are not present. This is due to the regularising effect which improves the accuracy of the SPH approximation. However, particle shifting like this does violate momentum conservation (though the discrepancies are very small), and so ideally particle-shifting should only be used when necessary and with the smallest possible coefficient that still produces desirable results.

2.6.4 XSPH

A further technique for maintaining particle regularity, is to update the particle position according not only to velocity solved from the momentum equation, but also to the local average of the velocity. This is known as the XSPH method, first proposed by Monaghan

[113]. The positions of particles are then updated following

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{u}_i - \epsilon \sum_j V_j (\mathbf{u}_i - \mathbf{u}_j) W_{ij} \quad (47)$$

where ϵ is an adjustable parameter usually set between $0.3 \sim 0.5$ depending on the desired strength of the effect. The XSPH method acts to partially move particles with a local smoothed velocity rather than just the velocity calculated at each individual particle. It is noted that the velocity itself is not updated, only the positions of particles, thus acting as a particle regularisation technique. Setting $\epsilon = 0$ returns the traditional SPH model whilst $\epsilon = 1.0$ moves particles solely based on the smoothed velocity field. Moving particles with the smoothed velocity acts to reduce disorder at short length scales by maintaining a consistent local velocity which allows particles to main a consistent separation. The method conserves linear and angular momentum, but not energy [116]. Some results have found that under certain conditions, XSPH can lead to reduced accuracy and increased disorder [143], and as such should be used with caution. In the current implementation XSPH is employed, though with a smaller value of $\epsilon \sim 0.2$, as this seems an ideal approach to improve small scale regularisation whilst avoiding the dynamics of the problem being affected.

2.7 Computational implementation

Having discussed the mathematical formulation and numerical techniques of the method, how these fit into the structure of an SPH code will now be briefly addressed. Here we will discuss how an SPH code works, including time stepping, the overall computational scheme of the method, and how the SPH method can be effectively scaled through the use of parallel programming. The specifics of the current implementation will be the main focus.

The SPH code used in this work was developed on top of the code base developed by Professor Stephen Neethling at Imperial College London. This is a highly modular C++ code, utilising large scale parallelisation with the message passing interface (MPI) specification. This was done primarily for use on multiple node, distributed-memory systems such as high performance computing clusters. The original code contained many standard, but key, implementations before work was started. This included neighbour listing and searching, domain subdivision across multiple processors, message and data passing, particle interaction loops, and particle based boundary methods. Of course in this work any additions to the code had to be made within the existing framework, and so it was always necessary to consider the computational loop, and how data was stored and shared, particularly in the context of an MPI implementation.

Many details of this specific C++ SPH implementation can be found in a publication of Professor Stephen Neethling [122] and the PhD thesis of one of his previous students, Dr Dan Barker [13]. The focus in this work was on the development of additional and improved numerical methods and their understanding, rather than the computational implementation of an SPH code from the ground up. In the case where an SPH code must be implemented from scratch, there are many important factors to consider, and whilst these will not be addressed in-depth here, the basic ideas will now be presented. For further reading on the topic the reader is directed towards the appropriate literature [43, 45].

SPH codes can be simplified into three main steps: (i) neighbour listing, (ii) particle interactions, (iii) system update. This represents an implementation whereby the neighbouring particles within the smoothing kernel are found, the interactions between a particle and its neighbours are calculated according to the governing equations, and then the new time step is computed according to a time integration scheme and the relevant values are updated. The procedure is as follows:

(i) Neighbour listing - Cell-linked list method [43]

- The domain is divided into square cells of length equal to the support domain of the kernel (usually $2h$).
- Particles are listed according to the cell to which they belong; this gives a reordered list of particles.

(ii) Particle interactions

- Particles in the same cell and adjacent cells are searched - if they have a distance of less than $2h$ then they are a real neighbour.
- Interactions are calculated between a particle and its neighbours according to the governing equations.

(iii) System update

- A variable time step is computed.
- Quantities are updated according to the time-stepping scheme.

This process is then repeated for the desired run-time. During this process and at certain intervals it may be necessary to complete additional operations, such as saving data or running smoothing procedures.

Whilst calculating particle interactions, it is possible to exploit the symmetry of the interactions in order to optimise the process. For example, whilst searching the neighbours

of particle i we find particle j , and whilst searching the neighbours of particle j we find particle i . Rather than calculating the interaction between particle i and j twice, it is possible to choose an appropriate cell stencil such that each pair is only calculated once. The stencil means that calculations are not calculated over all neighbouring cells, but only neighbouring cells with a greater relative index. An example of such a neighbour stencil can be seen in Fig. 5. This process reduces the number of interactions calculated by half and so offers nearly a two-fold speed up. This is commonly implemented in most SPH codes [63] [45].

The memory locality and CPU architecture should also be considered when implementing an SPH code. When simulating very large numbers of particles, lots of data associated with the particles must be shared, accessed, and stored, repeatedly throughout the simulation. Small optimisations to these procedures based on data read/write operations can often produce large improvements in simulation speed. For example, structuring memory such that local particle data is stored contiguously and can be read and written within the cache [126]. Further, by leveraging Single Instruction, Multiple Data (SIMD) instruction sets available on CPUs, particle interaction loops can be vectorised to greatly improve speed [153].

José Domínguez has produced valuable literature studying these kinds of optimisations [45]. It is noted that whilst effective, these methods often involve restructuring and rewriting large parts of the code-base and so many are not implemented in the code used in this work where the focus was on the numerical techniques. In future work when the optimisation and speed of the code is addressed, this may be relevant.

2.7.1 Time integration schemes

Having detailed the spatial discretisations to produce governing equations for each particle, it remains to use these formulae to update the velocities, positions, and densities of each particle to progress the simulation forward. Of course, if a density summation is used as described by Eq. (28) in Sec. 2.6.1, then the density can be calculated at each time step independent of time, though this is not the formulation used in the current implementation.

There are two aspects that are important to time integration in SPH: the time-stepping scheme for updating values, and the stability criteria on a variable time step.

Time-stepping

SPH usually uses explicit time integration schemes to match the explicit weakly-compressible formulation and keep computational costs low. In order to strike a balance between accuracy and computational cost, usually a predictor-corrector or leapfrog scheme of $\mathcal{O}(\Delta t^2)$ is used. This matches the $\mathcal{O}(\Delta h^2)$ accuracy of the SPH scheme. The commonly-used 4th-order Runge

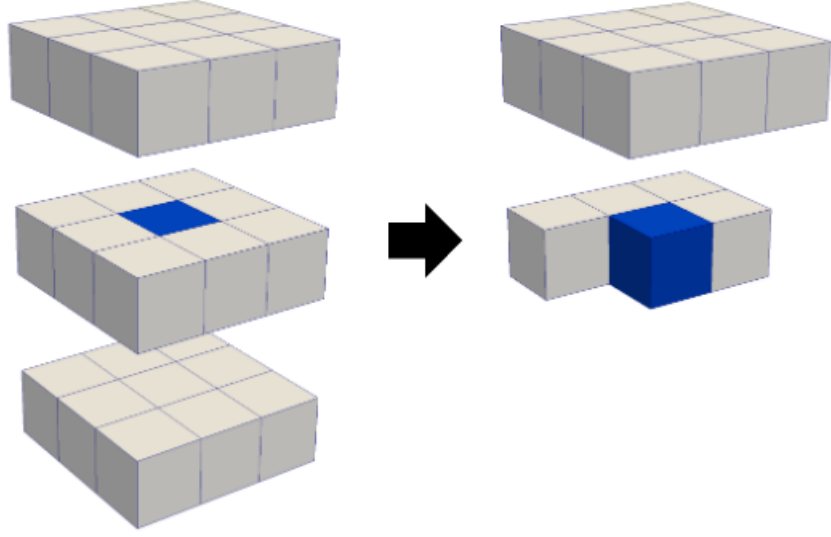


Figure 5: The neighbour cell stencil only calculates interactions over certain neighbouring cells in order to avoid double counting pairs of particles. Applying this stencil throughout the computational domain ensures each particle interaction is only counted once and can produce a nearly two-fold speed up.

Kutta (RK4) is sometimes seen, though this carries a high computational cost owing to the 4 intermediary stages of particle variables that must be calculated. Despite the higher order accuracy of RK4 compared to standard second-order methods it does not always produce more accurate results [97], and this is likely owing to some of the conservation properties associated with the second-order schemes which are often symplectic integrators.

The predictor-corrector scheme implemented here is as follows:

Predictor:

$$\begin{aligned} \mathbf{r}_i^{(t+\Delta t/2)} &= \mathbf{r}_i^{(t)} + \frac{\Delta t}{2} \mathbf{u}_i^{(t)} & \mathbf{u}_i^{(t+\Delta t/2)} &= \mathbf{u}_i^{(t)} + \frac{\Delta t}{2} \frac{d\mathbf{u}_i}{dt}^{(t)} \\ \rho_i^{(t+\Delta t/2)} &= \rho_i^{(t)} + \frac{\Delta t}{2} \frac{d\rho_i}{dt}^{(t)} \end{aligned}$$

Corrector:

$$\begin{aligned} \mathbf{r}_i^{(t+\Delta t)} &= \mathbf{r}_i^{(t)} + \Delta t \mathbf{u}_i^{(t+\Delta t/2)} & \mathbf{u}_i^{(t+\Delta t)} &= \mathbf{u}_i^{(t)} + \Delta t \frac{d\mathbf{u}_i}{dt}^{(t+\Delta t/2)} \\ \rho_i^{(t+\Delta t)} &= \rho_i^{(t)} + \Delta t \frac{d\rho_i}{dt}^{(t+\Delta t/2)} \end{aligned}$$

This requires two evaluations of time derivatives at particles - once for $\mathbf{a}_i^{(t)}$ and $d\rho_i/dt^{(t)}$

at the predictor step, and a second time for $\mathbf{a}_i^{(t+\Delta t/2)}$ and $d\rho_i/dt^{(t+\Delta t/2)}$ at the corrector step.

The corrector step can also be modified to

Corrector:

$$\begin{aligned}\mathbf{r}_i^{(t+\Delta t)} &= \mathbf{r}_i^{(t)} + \frac{\Delta t}{2} \left(\mathbf{u}_i^{(t)} + \mathbf{u}_i^{(t+\Delta t/2)} \right) & \mathbf{u}_i^{(t+\Delta t)} &= \mathbf{u}_i^{(t)} + \frac{\Delta t}{2} \left(\frac{d\mathbf{u}_i^{(t)}}{dt} + \frac{d\mathbf{u}_i^{(t+\Delta t/2)}}{dt} \right) \\ \rho_i^{(t+\Delta t)} &= \rho_i^{(t)} + \frac{\Delta t}{2} \left(\frac{d\rho_i^{(t)}}{dt} + \frac{d\rho_i^{(t+\Delta t/2)}}{dt} \right)\end{aligned}$$

where the predictor and corrector time derivatives are averaged for the final value. However, this requires the time derivatives to be stored at both the predictor and corrector steps and thus imposes additional memory requirements. In practice, either corrector step formulation was found to produce very similar results in the problems of interest.

Time-step criteria

The use of an explicit time integration scheme makes each time step computationally less expensive and more straightforward to implement, but does impose strict stability criteria on the time step itself. This is one of the main drawbacks of using a fully explicit scheme.

The usually relevant time-step criteria for SPH are: the CFL condition, the forcing/acceleration terms, and the viscous diffusion time step.

$$\text{CFL condition} \quad : \quad \Delta t_{\text{CFL}} \leq \min \left\{ \frac{h}{C_s} \right\} \quad (48a)$$

$$\text{Acceleration condition} \quad : \quad \Delta t_{\text{F}} \leq \min \left\{ \sqrt{\frac{h}{|\mathbf{a}_i|}} \right\} \quad (48b)$$

$$\text{Viscous diffusion condition} \quad : \quad \Delta t_{\nu} \leq \min \left\{ \frac{h^2}{\nu} \right\} \quad (48c)$$

It can firstly be noted the importance of the speed of sound C_s in the system. If a realistic speed of sound for the fluid is used then this will make the time step (48a) prohibitively small. It is for this reason that an artificial speed of sound is used, which is chosen to limit compressibility whilst also allowing reasonable time steps. The acceleration condition (48b) accounts for any accelerations that would be large enough to produce a velocity change which would then violate the CFL condition. The viscous diffusion time step (48c) is the usual condition on diffusion speed. In an SPH formulation that uses an artificial viscosity, then the artificial representation of ν is used in the condition (48c). In practice, the speed of sound CFL condition (48a) is the most limiting, especially for coastal wave problems where impact or gravity-induced pressure forces are dominant.

Other stability criteria that are sometimes included are the DEM rigid contact time scale which will be addressed in Sec. 6.2 and surface tension capillary forces if they are modelled.

2.7.2 Parallel framework

Computational runtime of an SPH implementation is largely tied to the number of particles in the system, and the runtime can dramatically increase at higher resolutions or larger scale systems. For this reason, efficient and scalable parallelisation of the code is essential.

SPH as a method lends itself naturally to parallelisation since it can be solved in a fully explicit framework. An explicit formulation means that at any given instant all particle variables are known and relatively independent. This makes it straightforward to assign particles to different processes and allow them to evolve individually with limited communication and latency.

Particle assignment to processes is usually done via domain decomposition. This involves splitting the simulation domain into subdomains, and allowing each process to handle a single subdomain. Since all particles need to interact with their neighbours within the radius of the smoothing kernel ($2h$), each process must share its outermost - halo - particles with the neighbouring processes via a data exchange at each time step. This is known as the halo exchange, and is common practice in SPH and many other parallel code implementations. For particles located in the cells at the edge of each subdomain, their halo will overlap into neighbouring process subdomains and as such their data must be sent and received appropriately. It is always important in an SPH implementation to be aware whether the correct data has been sent and received before each calculation. Data exchange between processes can potentially represent a communications bottleneck, and so it is important to minimise this data exchange whilst also minimising the workload on each processor. There is therefore usually an optimum number of processes that can be assigned to a simulation before parallelisation overheads reduce the simulation speed.

As opposed to a mesh-based method, the Lagrangian nature of SPH means that particles (nodes) are constantly moving, and especially in the case of free surface problems, different subdomains of the domain can host a very different number of particles from one time to another. Since the number of particles largely determines the computational runtime, it is important to load balance between the particles assigned to each process throughout the simulation. In general, load balancing is always desired to assure that each process has an equal amount of work to minimise process idle time and keep good synchronisation. This is done in SPH by periodically resizing the subdomains to ensure that each process handles a similar number of particles.

Oger et al. have published a comprehensive discussion of the parallelisation methods in

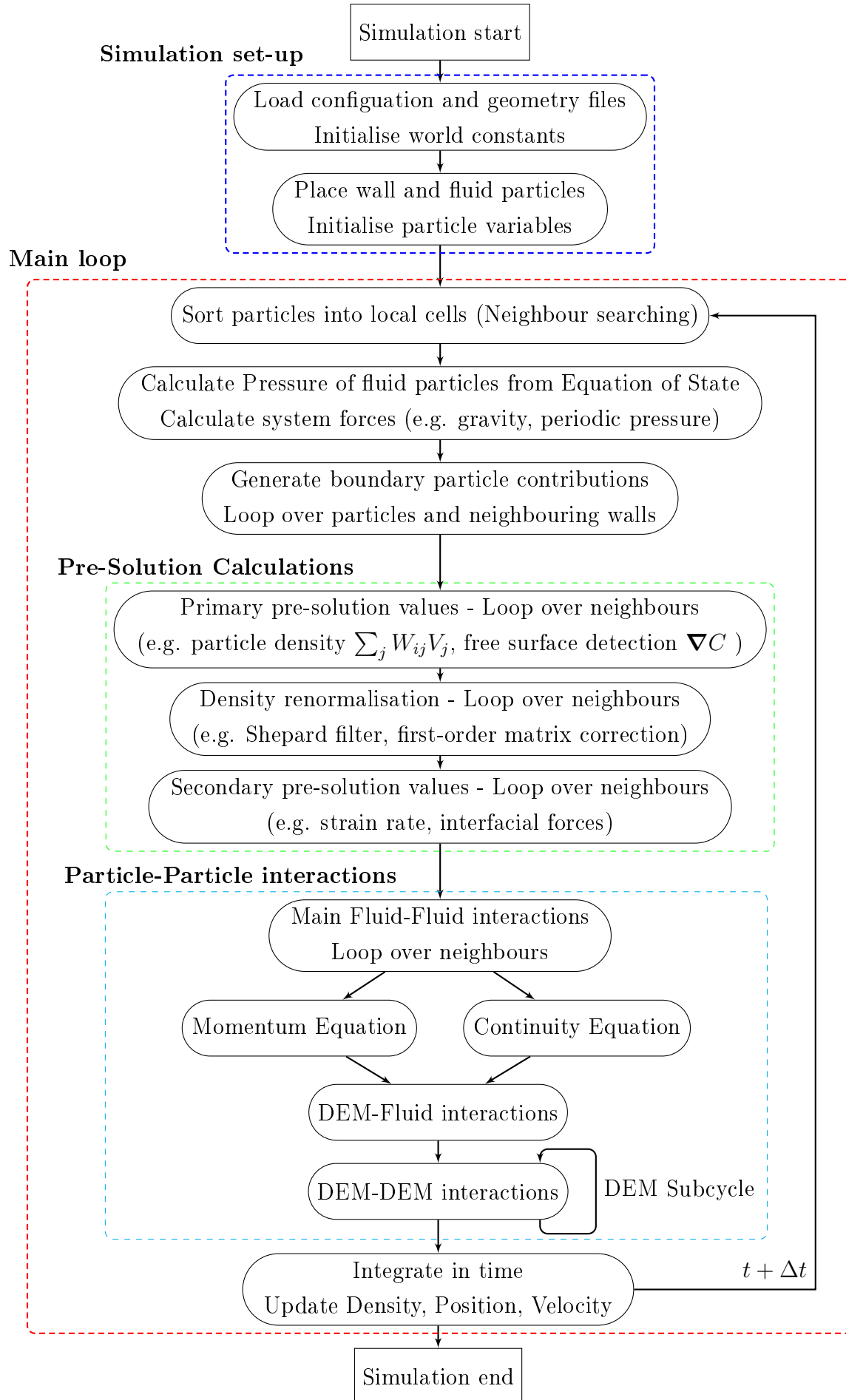
SPH, with algorithms and efficiencies for domain decomposition and load balancing, as well as overall performance of massively parallel MPI SPH implementation [127].

Aside from CPU-based parallelisation, a promising and now well-developed alternative for code acceleration are Graphical Processing Units (GPUs). GPUs offer the advantage of extremely efficient multi-threading capabilities and are ideal for numerical calculations which require a large throughput of data. If implemented and optimised effectively, a GPU executed code can be comparable with if not faster than typical large cluster machines. The full extent of an SPH GPU implementation contains many specifics and nuances and so will not be handled here, but the reader is pointed to available literature which handle this topic - [66, 34, 45].

With big advances in GPU capabilities and programming features, the implementation of SPH on GPUs has become highly optimised and wide-spread in the open-source codes. DualSPHysics [33], GPUSPH [69], and AquaGPUSPH [26] to name just a few of the more well-known options. DualSPHysics represents one of the more pioneering and effective GPU implementations, being very well documented, actively developed, and achieving speed-ups of almost two-orders of magnitude when compared to CPU implementations [34]. The code presented in this work is implemented in a massively parallel distributed memory framework, ideal for running on a large number of CPUs on a high performance cluster. Rewriting the code-base to be functional in a GPU framework would be a logical next step to greatly improving simulation speeds but was beyond the scope of this work.

2.7.3 Computational algorithm

A brief discussion of the computational implementation of an SPH code was presented at the beginning of this section. Here, the outline of the computational algorithm for the SPH code used in this work is presented.



It is notable that most, if not all, of the “*Pre-Solution Calculations*” can be avoided if not employing certain numerical procedures, saving a large amount of computational expense. In this work, the only pre-solution calculations carried out are those required for free surface detection when using particle shifting (Sec. 2.6.3), though many other SPH implementations will calculate further variables required before the main fluid-fluid interactions during these steps. The steps “*Generate boundary particle contributions*”, “*DEM-Fluid interactions*” and “*DEM-DEM interactions*” are important steps which have not yet been addressed, but will be discussed in detail in the following Chapters 3 and 6.

3 Wall boundaries in SPH

As previously discussed, wall boundaries in SPH remain an open problem and are even noted as one of the grand challenges facing development of the method [160]. The main issue in handling solid boundaries for SPH is the truncation of the kernel support domain due to the boundary. The derivations for the kernel approximation in SPH presented in Sec. 2.3 only remain accurate to $\mathcal{O}(\Delta x^2)$ as long as the kernel has a complete support domain and remains well sampled by integration points throughout. The presence of a wall boundary presents regions where fluid particles may not enter and thus leads to an incomplete kernel support. A schematic diagram of this is shown in Fig. 6. Further, any boundary method must of course accurately emulate a real solid boundary, achieved by satisfying the appropriate boundary conditions. This means the method must provide adequate and accurate repulsion at the boundaries, whilst still allowing the fluid particles to flow along and up to the walls in a physically realistic manner.

Regardless of the choice of boundary method, wall boundaries in SPH are almost always discretised into *boundary* particles. These are essentially fluid particles which remain fixed at a location along the boundary of the wall but do not follow the fluid equations of motion. Typically their variables (velocity and pressure/density) are calculated differently depending on the chosen boundary method or desired characteristics. These will be discussed in the following sections. Boundary particles are particularly useful in the implementation of many of the boundary methods. They allow derivatives near walls to be calculated using one of the more accurate formulations in Eqs (19) and (20) which rely on multi-particle values (i.e. values at the fluid and the wall). Boundary particles can also be useful for tracking values on or near walls.

Typically, there have been three different approaches to model solid boundaries in SPH: Repulsive functions, ghost boundary particles, and completing the governing equations by specifically including the boundary terms - so-called semi-analytical boundaries. The choice of boundary method is often a very important factor in an SPH simulation [35]. In this section, each of these methods will be discussed, with some further discussion on their general efficacy and implementation which is not always addressed in the literature.

The following chapters will then present the theory and implementation of improvements to the semi-analytical boundary method, with aims to make the method more accessible to the wider research community and more viable at a simple implementation level. Some example simulations will be presented focusing on the accuracy and improvement of the semi-analytical boundary method. The benchmark of choice will be the 3D dam break -

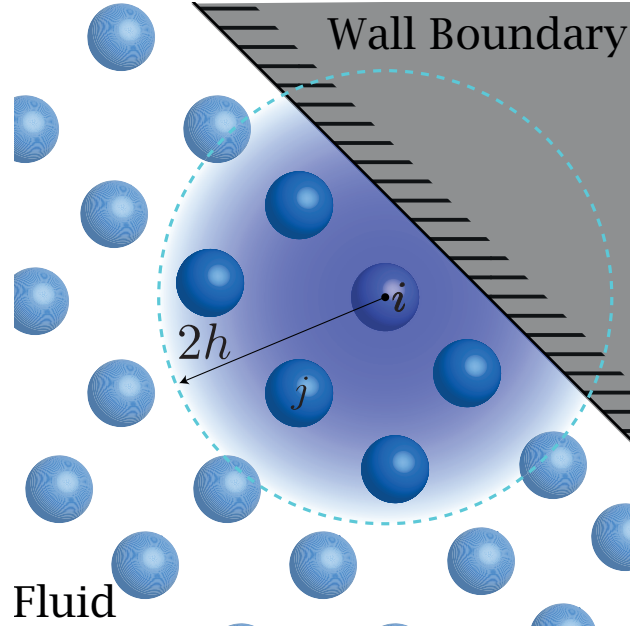


Figure 6: Truncated kernel support domain near a solid boundary. The support domain of the i th fluid particle is highlighted by the blue shaded region and circular dashed line. As can be seen, in regions where the wall boundary intersects the kernel support, there is a deficiency due to the lack of fluid particles within the solid boundary domain.

one of the standard validation tests suggested by SPHERIC [149]. This benchmark contains clear experiment criteria and good quality data for both experimental and numerical results. The original experiment was carried out by Kleefsman [81]. In this work, the dam break will be used as one of the main benchmarks for assessing the quality of different boundary methods. In particular, the overall dynamics and the behaviour of the pressure near the fluid-boundary impact points will be assessed, as these are typically a good proxy for an effective boundary method. Further, the 3D dam break simulates the dynamics of interest for coastal problems, involving breaking waves and high-speed impacts of fluid against a boundary.

3.1 Repulsive containment and leaked particles

The simplest approach to handling boundaries is to have *repulsive walls*. Near-wall fluid particles are identified and prescribed an additional repulsion force based on their normal distance to the wall. The motivation behind this boundary method is to reflect the fundamental nature of the fluid containment due to a solid boundary, which is the intermolecular forces that prevent penetration of the two mediums. For this reason, the force usually has the form of a Lennard-Jones potential between the fluid particles and boundary particles, which was first suggested by Monaghan [112]. However, given that SPH as a method does

not attempt to model the molecular behaviours in either scale or mechanics, this motivation is a purely theoretical concept, and in practice the repulsive boundary method is purely numerical with only very vague physical motivation.

Whilst very good for containment of particles and very simple to implement, the simple methodology suffers from a few issues. Firstly, the boundary cannot simulate shear stress since it can only produce forces normal to the wall which behave purely repulsively. This means only full-slip conditions can be emulated. Most importantly, introducing a repulsive wall force will break momentum conservation. The formulation lacks physical rigour and is effectively a purely numerical addition, meaning the relevant parameters in the Lennard-Jones type equation are unclear. Setting these parameters in SPH is not fully studied, often requiring different values for different problem types [12].

For these reasons, repulsive forces are very rarely used as a method for modelling boundaries in modern SPH codes. That being said, they can be used to provide a straightforward and effective way to limit particles which leak into solid boundaries. In any SPH simulation, if a fluid particle approaches a solid boundary at high enough speeds, it is a possibility that the chosen boundary method cannot provide adequate repulsion to prevent fluid penetration. This is especially relevant at complex solid boundaries where multiple boundary edges connect and the geometry may not be well sampled at the given resolution. In these cases, repulsive forces can be used as a supplement to another boundary method in order to limit particle leaking.

“Leaked” fluid particles present problems in the computation as their behaviour is not well-defined. Of course, the fluid should never penetrate an impermeable solid boundary, and in the case of a large number of leaked particles it would suggest an inadequate boundary method. However, the nature of weakly compressible SPH means it is expected that individual fluid particles can sometimes exhibit large spikes in their pressures and velocities without affecting the overall dynamics. These particles then carry the risk of being able to penetrate into solid boundary regions and “leak” from the fluid domain.

Leaked particles are not generally addressed in the literature. This may be because they are a rare occurrence, only happening in simulations which involve high velocity fluid impacts against complex boundaries, but as explained, they are a distinct possibility. As such, any SPH code should contain methods for handling them. Perhaps the most straightforward approach is to remove particles which have leaked - the approach for example used by DualSPHysics [42]. This should be carefully done by appropriately deleting the particle from memory and removing it from the linked lists involved in calculating particle neighbours. One slight issue with this, is that removed particles will then of course leave voids behind near the point they impacted the boundary. As long as leaked particles are a rare occurrence, this

usually does not cause any effect on the overall flow dynamics. An alternative method which can help mitigate this issue, is to reinsert leaked particles back into the fluid domain. This can be done by reflecting particles back over the boundary face through which they leaked and assigning them a velocity and pressure based on a sum over the local fluid particles (as in Eq. (2)). Of course, this is not necessarily physically consistent, and may have conservation issues, but it does ensure that the number of particles in a simulation remains constant and that certain near-boundary regions maintain a well-sampled particle support.

As will be seen later in this section, some boundary methods are more prone to leaked particles, to such an extent that flow dynamics may be affected. For this reason, utilising repulsive forces as a secondary or back-up containment method can be considered. Ultimately, for all the results presented in this work, repulsive containment forces were not deemed necessary. Early iterations of the semi-analytical boundary method (which will be discussed in Sec. 3.3) were prone to large numbers of leaked particles and hence motivated the work here. However, the improved final version presented in Secs 4 and 5 did not suffer from the same issue, displaying greater containment and thus not needing any repulsive boundary forces. Nonetheless, the work is still presented here in case it is of value to a future implementation.

The activation of repulsive forces will in most instances not be required and so not affect the flow dynamics. In instances where the forces do activate, the formulation should have a physical underpinning and be as small as necessary to avoid particle leaking. The standard formulation of repulsive forces was not considered appropriate for this use, and so a new formulation was written in order to improve its behaviour as a fluid containment method. Firstly, rather than only generate repulsive forces between fluid particles and local wall particles, repulsive forces are applied to fluid particles based on their distance to the nearest wall independent of the placement of boundary particles. This allows the forces to be generated consistently near walls rather than relying on a boundary particle discretisation. Since containment issues can often occur near boundary intersections where the walls are not well sampled by boundary particles, it is not as effective to rely on said particles to generate the repulsion. Instead, the forces are generated regardless of the presence of boundary particles. This does incur some overhead in the computations, since the forces are not generated between particles, they cannot be done as a cheap additional calculation in the main particle-particle interactions. Instead, fluid particles must first be marked as being near a wall whilst boundary contributions are generated (*“Generate boundary particle contributions”* step in 2.7.3). Then, the repulsive force is calculated by looping over each marked particle and its nearby walls, typically before the main particle-particle loop.

In an attempt to make the new formulation more physically consistent, it now generates a physically derived energy on the approach of a particle close enough to a boundary. This

formulation of the repulsion mechanism relies not only on distance to the wall, but also on normal velocity flux at the wall. For a particle i within a certain repulsion range r_0 of a wall, the pressure and flux style contributions are scaled by

$$\begin{aligned}\Phi_i^P &= P(\rho' = 1 + \xi_P) \\ \Phi_i^v &= \xi_v \frac{1}{2} \rho_i (\mathbf{v}_{ib} \cdot \mathbf{n})^2\end{aligned}\tag{49}$$

Φ_i^P and Φ_i^v encode a compression contribution and normal velocity flux contribution to a pressure style repulsion respectively. Together they form a representation of the energy per unit volume generated by the repulsive force mechanism. ξ_P is a user-defined constant which controls the strength of the compression repulsion. This is done by calculating a pressure using the equation of state (Eq. (30)) with a relative density of $\rho' = 1 + \xi_P$ to generate an additional positive pressure force near the boundary (i.e. $\xi_P = 0.001$ scales the pressure force equivalent to a compression of 0.1%). ξ_v is also a user-defined constant, which is used to control the strength of the normal velocity flux contribution to repulsive pressure. $\mathbf{n}$ is the inward normal of the wall, and $\mathbf{v}_{ib} = \mathbf{v}_i - \mathbf{v}_b$, where $\mathbf{v}_b$ is the velocity of the wall boundary (most often $\mathbf{v}_b = 0$). Varying ξ_P and ξ_v allows the strength of the distance and velocity effects respectively to be controlled. In general, using these repulsion mechanisms as a backup containment method suggests Φ_v should be the dominant factor, as it is usually high velocity particles which are prone to leaking and this means the forces are only produced on particles which really need it.

Eqs (49) present a typical scale of the injected energy for nearly leaking particles. Of course, this must be scaled by some form of the distance to the wall in order that particles feels a smooth and monotonically increasing repulsive force. For this, the classic Lennard-Jones style scaling used.

$$\begin{aligned}q &= \max\left(\frac{|\mathbf{r}_{ib}|}{r_0}, d_{\min}\right) \\ \mathbf{D}_i^{LJ} &= \left(\frac{1}{q^4} - \frac{1}{q^2}\right) \frac{\hat{\mathbf{r}}_{ib}}{qr_0}\end{aligned}\tag{50}$$

Here $\mathbf{D}_i^{LJ}$ is the scaling function, and r_0 is the repulsion range. This is the distance from the wall at which the repulsive containment is activated, and should typically be set as small as possible such that the force is only applied to particles which are very near to leaking. The value of r_0 will vary depending on the boundary method used, since different boundary methods may have a different mean distance at which particles settle near the wall. $\mathbf{r}_{ib}$ is the vector pointing from particle i to the nearest point on the wall boundary from which the repulsive force is generated, and $\hat{\mathbf{r}}_{ib}$ is the respective unit vector identifying the direction

from the particle to the wall. d_{min} is a constant which controls the distance at which the repulsive force reaches a maximum, and is set as a fraction of r_0 . The use of d_{min} as a limiter ensures that the repulsive force does not spike too much very close to the wall due to the exponential behaviour of $\mathbf{D}_i^{LJ}$.

Finally, the force is calculated as

$$\mathbf{F}_i^R = \begin{cases} -(\Phi_i^P + \Phi_i^v) \mathbf{D}_i^{LJ} & \text{if } |\mathbf{r}_{ib}| < r_0 \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (51)$$

It is noted that this formulation of repulsive forces contains multiple constants that are user-defined and may be problem dependent - $r_0, \xi_P, \xi_v, d_{min}$. This can of course be an additional difficulty for implementation of the method. However, r_0 and d_{min} can be defined on a per-boundary method basis and are usually fairly trivial to assign as will be seen in following sections, and whilst ξ_P and ξ_v are problem dependent, their physical intuition is very straightforward as just the strengths of the repulsion mechanisms, and so they are quite simple to adjust based on problem requirements.

3.2 Ghost particles

The most commonly implemented boundary methods rely on so-called “ghost” particles being inserted behind the wall boundaries to mimic the presence of a solid boundary. The particles behave as “ghost” particles in the sense that they provide a wall-like presence to the fluid particles but do not themselves interact with each other nor are they free to follow equations of motion. These particles will be here referred to as *wall* particles, and belong to a similar class of particles as the previously mentioned *boundary* particles. Wall particles, however, are placed behind the wall boundary and are given tailored properties and field values such that their presence mimics the effect of a solid wall boundary. They ensure that SPH summations and calculations remain well-sampled near a boundary, and also provide a restoring force near the walls which helps fluid particles remain contained. A schematic diagram of their presence can be seen in Fig. 7.

In literature the distinction between *boundary* and *wall* particles is not always addressed, and often times it is trivial and logical to treat them as one in the same. When discussing the different available boundary methods for SPH however, it is often useful to make the distinction. From hereon, “wall particles” will be used to generically refer to any wall-discretising ghost particle except when otherwise noted.

Containment mechanism

The mechanism of the containing force of ghost particles is implicitly produced, and is as

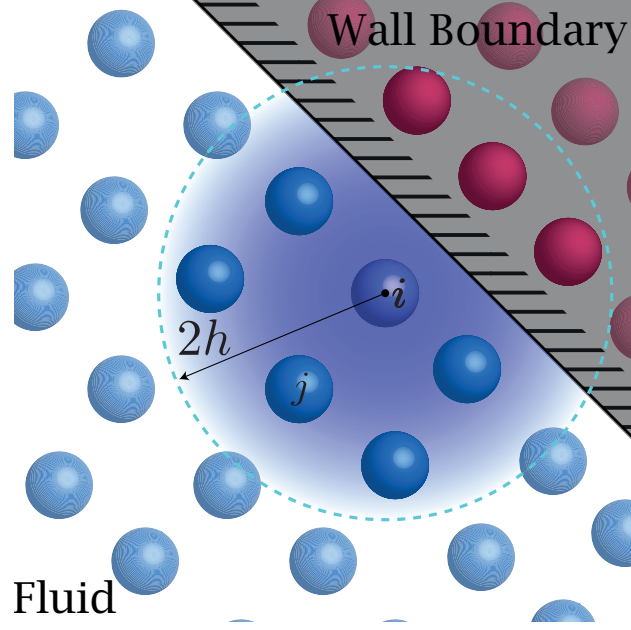


Figure 7: Kernel support domain near a solid boundary containing boundary particles. The support domain of the i th fluid particle is highlighted by the blue shaded region and circular dashed line. As can be seen, in regions where the wall boundary intersects the kernel support, the red wall particles within the solid boundary domain ensure there is no deficiency near the boundary. The previously described *boundary* particles are not depicted in this image.

follows:

1. Fluid particles which approach a boundary experience an increase in density due to the stationary ghost particles, following the continuity equation (Eq. (36)).
2. The density increase results in a local pressure increase, following the equation of state (Eq. (30)).
3. The increase in local pressure leads to an increase in the pressure term in the momentum equation (Eq. (27)).
4. The pressure term increase points away from the ghost particle and thus generates a repulsive acceleration.

This mechanism was first discussed in the literature by Crespo et al. [2].

The issue with relying on this style of repulsive force mechanism is that it may lead to an unphysical gap between the boundary and where the fluid particles lie. Specifically, when fluid particles flow into a dry boundary region, they are held at a unphysical distance from the wall due to the repulsive mechanism, creating a noticeable “gap”. This phenomenon is generally well-known, particularly to the users of the very popular open-source SPH code *DualSPHysics* which utilises ghost particles in the *dynamic boundary condition* [33, 2].

Boundary particle positioning

An approach to reduce the presence of the unphysical boundary gap is to place the ghost wall particles offset from the actual physical wall boundary position. In this way, their presence is only felt closer to the actual wall position they are trying to emulate and can produce a more accurate near-wall behaviour. For example, the *dynamic boundary condition* is typically seen to produce a gap on the order of Δx , and so if the wall particles are offset into the walls by a distance of Δx , the fluid is able to flow along the physical wall boundary with no gap. The downside of this is that it greatly increases the chances of fluid particles leaking, or penetrating the solid wall domain, which is often undefined behaviour. This issue has been addressed in Sec. 3.1. If this is to be done accurately, the size of the gap should be recorded with some test simulations and then used to inform an appropriate wall particle offset. Typically the size of the gap will be consistent for the boundary method chosen, and so it can be then continually used if desired.

Setting variables

By appropriately setting the variables of boundary and wall particles, their presence can also be used to help the pressure and velocity fields satisfy appropriate boundary conditions, or even help modulate other desirable behaviours at the boundary.

In the case of the *dynamic boundary condition* used in *DualSPHysics*, the density and pressure of the wall particles are updated according to the continuity equation (Eq. (26)) and equation of state exactly as fluid particles, but are otherwise held fixed within the wall. This has the benefit of making wall particles trivial to include in the regular fluid particle loops which can greatly help with computational speed and implementation. A recent improvement to this method dubbed the *modified dynamic boundary condition* calculates the ghost particle pressure from a first-order accurate extrapolation from the fluid domain. This is a more involved process, but was shown to reduce the issue of the unphysical boundary gap and greatly increase accuracy of results near the boundary whilst only increasing computational expense by 10% – 20% [49].

A second option, is to set a pressure value at the wall, P_w , according to local fluid pressure values, P_f , in order to satisfy a Neumann or Dirichlet boundary condition. The density at the boundary/wall particle can then be found by an inversion of the equation of state Eq. (30).

Most often, the Neumann boundary condition $\partial P / \partial n = 0$ is set, though in the presence of gravity this should more appropriately be set as $\partial P / \partial n = \rho \mathbf{g} \cdot \mathbf{n}$. Integrating along the centreline of a fluid-boundary particle pair would lead to a form of the pressure at the boundary particles being calculated as

$$P_w = P_f + \rho_f (\mathbf{g} - \mathbf{a}_w) \cdot \mathbf{r}_{wf} \quad (52)$$

where for completeness, $\mathbf{a}_w$ is the acceleration of the wall boundary (often just $\mathbf{0}$), and $\mathbf{r}_{wf} = \mathbf{r}_w - \mathbf{r}_f$ is the position vector pointing from the fluid to the wall particle under consideration. This form of calculating pressure at the wall can be seen in the work of Adami et al. [3], and is applied specifically to wall particles. Eq. (52) appropriately written in an SPH method should of course calculate the pressure at the ghost boundary particles based on a summation over the local fluid particles. This is written as

$$P_w = \frac{1}{\sum_f V_f W_{wf}} \sum_j V_f (P_f + \rho_f (\mathbf{g} - \mathbf{a}_w) \cdot \mathbf{r}_{wf}) W_{wf} \quad (53)$$

A Shepard Filter style summation as in Eq. (29) is used to extrapolate the pressure field to the boundary particles.

An adjustment to this form of boundary pressure, though specifically relevant to boundary particles, can be made by considering the motion of the fluid normal to the wall. In this case, by projecting the momentum euler equation in the direction normal to the wall, we have the more complete Neumann boundary condition $\partial P / \partial n = \rho (\mathbf{g} \cdot \mathbf{n} - \mathbf{a}_f \cdot \mathbf{n})$, where $\mathbf{a}_f$ refers to the acceleration of the fluid in the vicinity of the boundary. De Lefle [38] and Marrone [100] have both used this form of pressure boundary condition, and suggest an approximation for the acceleration of the fluid near the boundary based on the impermeability condition at the wall. Considering the relative normal velocity of the fluid to the local boundary $(\mathbf{u}_f - \mathbf{u}_b) \cdot \mathbf{n}_b$, and the reflection of a wave signal travelling at the speed of sound c_0 , they express the acceleration as

$$\mathbf{a}_f \cdot \mathbf{n}_b = -\frac{c_0(\mathbf{u}_f \cdot \mathbf{n}_b)}{(\mathbf{r}_b - \mathbf{r}_f) \cdot \mathbf{n}_b} \quad (54)$$

where $\mathbf{r}_b$ is the position vector of the boundary particle under consideration, and the subscript b is used to refer to properties defined at the given boundary particle. Combining this with a Taylor expansion of the wall pressure in both the tangential and normal directions [100], along with a suitable SPH extension, the formula to calculate the pressure at a boundary particle can then be given as

$$P_w = \frac{1}{\sum_f V_f W_{bf}} \sum_f V_f (P_f + \rho_f c_0 (\mathbf{u}_{fb} \cdot \mathbf{n}_b) + \rho_f (\mathbf{g} - \mathbf{a}_w) \cdot \mathbf{r}_{wf}) W_{wf} \quad (55)$$

Here there are presented three options for pressure assignment at the boundary: density values according to the continuity equation (26) and pressure values calculated from the resulting equation of state (30); a simple interpolation of the pressure based on local fluid

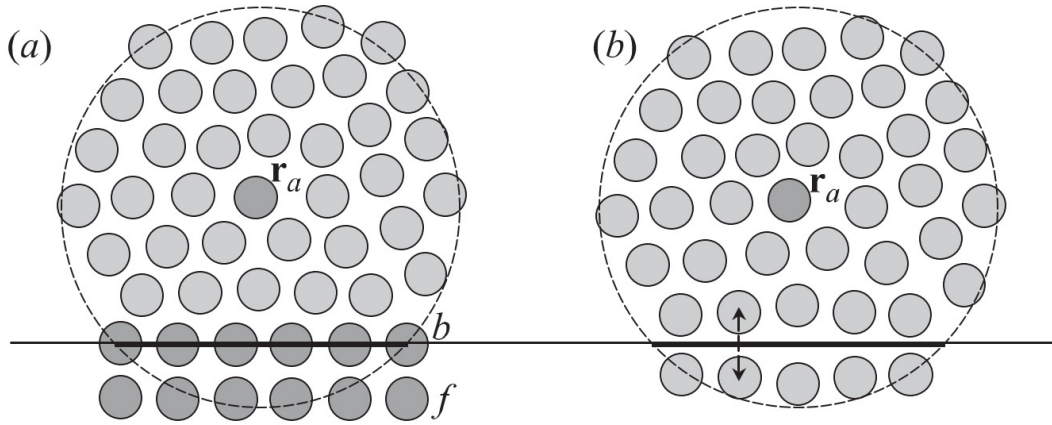


Figure 8: [161] Diagram of two wall particle methods: fixed - (a), and reflected - (b). In the fixed boundary particle method, particles in the wall can be distinguished into wall particles b that lie along the edge of the wall, and boundary particles f which lie within the boundary. Reflected particles are generated symmetrically to nearby fluid particles.

particles and the hydrostatic contribution due to gravity (Eq. (53)); and the same, but also including the contribution due to the local fluid acceleration (Eq. (55)). Different SPH implementations may make a different choice of pressure assignment depending on the specific boundary implementation employed, as well as other factors. These options will be briefly compared and the choice of which to use addressed in the following sections (Secs 3.2.1, 3.2.2, 3.3). Notably, however, the pressure assignment involving the wall-normal fluid flux (Eq. (55)) is generally not employed in the literature or available SPH implementations, and only addressed in the referenced papers [38, 100].

Types of ghost particles

Boundary particles are generally implemented in one of two ways: they are reflected, or mirrored, from fluid particles near the boundary at each time step and thus dynamically placed, or they are generated on a fixed mesh at simulation start and held static. These are here respectively known as *reflected* and *fixed* boundary particles.⁴ A diagram visualising the two methods is shown in Fig. 8.

Both methods ensure that near-wall fluid particles have sufficient support and attempt in some manner to satisfy the appropriate boundary conditions. Reflected particles aim to generate a minimum required number of boundary particles at the expense of re-generating them at each time step, whilst fixed particles need only generate the boundary particles

⁴In some literature, “ghost” particles are only used to refer to the *reflected* type boundary particles, perhaps due to their transient nature. In these cases, reflected and ghost particles may be referred to synonymously, while *fixed* boundary particles are addressed separately.

once, but often produce far more boundary particles than are needed at any instant. Of course, boundary particles are only necessary in regions near boundaries when we need to calculate fluid values in that region. Further, reflected particles can have assigned velocities, whereas fixed particles may only take the velocity of the wall in which they are embedded (often $\mathbf{v}_b = \mathbf{0}$).

We will briefly address both methods here.

3.2.1 Reflected particles

For reflected particles, any fluid particle near the solid boundary is “mirrored” into a ghost particle outside of the boundary. These reflected particles are re-generated at each iteration, and provided with parameters that are symmetric to those of the fluid particle they were mirrored from. This acts as a dynamic method for allocating boundary particles in regions where they are required. Sometimes the reflection process may rely on the presence of wall particles to generate the correct positions of the reflected ghost particles.

The reflection process near flat, convex boundaries is straightforward, but in non-convex boundaries near corners, the reflection process becomes ill-defined and requires more careful handling. In three dimensions the problem complicates even further, potentially with certain boundaries with intractable solutions.

In fact, the main constraint in using dynamically reflected particles is the difficulty in extending the method to 3D for complex geometries, particularly with sharp changes in the boundary. The reflection process can become unclear through multiple local faces [117], and the resulting ghost particles do not always adequately fill the space. This may lead to fluid particles with an undersampled support near complex boundary edges.

The problem of producing a consistent and effective boundary particle reflection process for all boundary types is not addressed in this work. Results in this work which use a reflected particles boundary method rely on a pre-existing algorithm in the in-house SPH code provided by Professor Stephen Neethling, details of which can be found in [13].

A recent improvement to the method of dynamically generating boundary particles was proposed by Fourtakas et al. [58] who used a local uniform stencil for each fluid particle. This method generates ghost boundary particles according to a stencil, which tracks each individual fluid particle, and is better able to handle complex geometries in 3D, nor relies on the presence of wall particles.

As reflected particles are dynamically generated, they may take any assigned velocity value. This allows certain velocity boundary conditions to be satisfied, in particular the no-slip and free-slip boundary conditions. This can be seen in Fig. 9, where mirroring the velocity allows an approximated null velocity field at the wall, satisfying the no-slip condition.

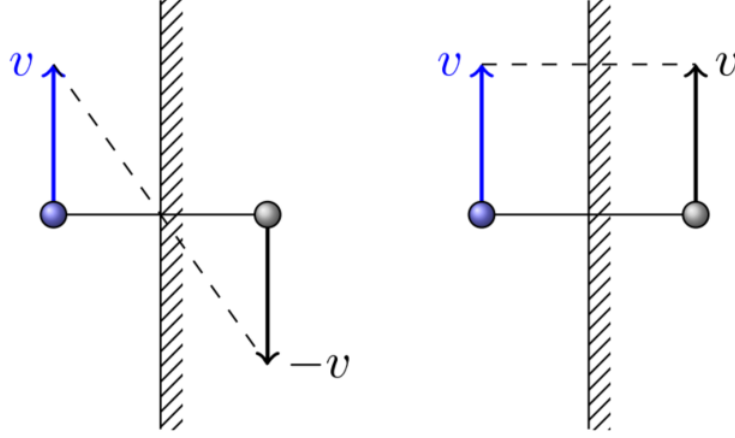


Figure 9: [13] By setting the velocity of reflected boundary particles appropriately, it is possible to satisfy either the no-slip or free-slip conditions at the wall as required.

Alternatively copying the velocity across the boundary approximates no tangential stresses at the wall for the free-slip condition.

In this work, the reflected particle boundary method is assumed the default implementation due to its speed, simplicity, and overall acceptable results. For any simulations presented in this work where the boundary method is not specified (nor the focus), it can be assumed that the reflected particle method was used.

3.2.2 Fixed particles

Fixed boundary particles are comparatively straightforward to implement, requiring only a one-time, grid-style placement which follows the boundaries of the domain. This can also be thought of as discretizing the boundary regions with fixed fluid particles. The boundary particles should be placed respecting the same initial spacing (resolution) as the fluid particles, and the same initial mass/density. Care should also be taken near corners to ensure the region behind them is filled in a consistent matter. This aspect is a typically much more tractable problem than the reflected particle equivalent.⁵

Fixed boundary particles need only extend out to a depth of $2h$ so that fluid particles near the boundary edge remain well sampled. To minimise the presence of an unphysical gap between the approaching fluid and the solid wall, the fixed boundary particles can also be placed offset into the wall by an appropriate distance.

Since fixed boundary particles are placed only once at the start of simulation and can have their values quite simply updated in a similar method as fluid particles, they are easily

⁵Results in this work which use a fixed particles boundary method rely on a pre-existing fixed particle placing algorithm provided in the in-house SPH code by Professor Stephen Neethling.

included in an extension of the fluid particle loops and can offer improved implementation and computational speeds, as seen in the *dynamic boundary condition* in *DualSPHysics*. Some of their speed benefit is lessened, due to the fact that they discretise the entire wall domain even when no fluid is near (unlike the reflected particle method), and so the method does tend to produce far more particles than are needed. The *dynamic boundary condition* or its modification are an example of fixed boundary particle methods.

As fixed particles cannot have variable velocities, inherently they cannot satisfy the no-slip condition. Their fixed velocity profile also means they tend to be less accurate in calculating derivatives at the boundary. Macia et al. produced a theoretical analysis on the accuracy of different ghost particle methods for emulating no-slip boundary conditions [98], and found that owing to invariable velocity, the fixed particle method was generally less accurate than the reflected particle method. Fixed velocity particles notably had a divergent second derivative in linear fields. In general, this makes the fixed particle boundary method undesirable for the purpose of accuracy, and its main benefits come from its ease of implementation and ability to handle complex geometries.

3.3 Semi-analytical boundary method

The boundary methods discussed so far rely on additions to the SPH method in order to replicate the effect of a solid wall boundary. The semi-analytical boundary method by contrast, aims to solve the original governing equations in a more complete manner, accounting for the presence of boundaries to the fluid domain and thus implicitly modelling their effect.

The original SPH integral approximation presented in Eq. (5) was accurate to second order given the two conditions in Eqs (7) - normalisation to unity and null first-order moment. These conditions do not hold however when close to a boundary since the support of the kernel function is truncated and thus incomplete. Accuracy can be restored near to a boundary by noting that the original formulation collapses to

$$A_I(\mathbf{r}) = A(\mathbf{r}) \int_{\Omega} W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' + \mathcal{O}(h) \quad (56)$$

and can then be rewritten to

$$A_I^\gamma(\mathbf{r}) := \frac{A_I(\mathbf{r})}{\gamma} = A(\mathbf{r}) + \mathcal{O}(h) \quad (57)$$

where

$$\gamma := \int_{\Omega} W(\mathbf{r} - \mathbf{r}') d\mathbf{r}' \quad (58)$$

Here, γ is effectively a normalisation term that restores first-order accuracy near a boundary by rescaling the value of the variable when the kernel support is incomplete. This normalisation corrects for the lack of normalisation to unity, but the first-order moment of the kernel remains non-null, and as such the accuracy of the approximation remains at $\mathcal{O}(h)$. It should be noted here that $\gamma = \gamma(\mathbf{r})$, and is indeed variable with position. As such, it is often defined at particle positions, and will often be written as γ_i or γ_a (γ evaluated from the particle at position $\mathbf{r}_i$ or $\mathbf{r}_a$ respectively). The γ superscript identifies a renormalised, or semi-analytical appropriate, version of an SPH approximation.

The main difficulty in the semi-analytical boundary method is the calculation of this normalisation term γ , which is addressed in its entirety in Chapter 4.

Far from the boundary $\gamma = 1$ as was assumed in the original derivation, and the original SPH approximation is restored with accuracy of $\mathcal{O}(h^2)$.

Noting this new form of normalised SPH approximation, it is necessary to re-derive the following discrete and derivative SPH approximations. The discrete operator is first written $A_D(\mathbf{r})$ (Eq. (2)), giving

$$A_D^\gamma(\mathbf{r}) := \frac{A_D(\mathbf{r})}{\gamma(\mathbf{r})} = \frac{A_i}{\gamma_i} = \frac{1}{\gamma_i} \sum_j A_j W_{ij} V_j \quad (59)$$

which is again, first-order accurate near a boundary, but with the original second-order accuracy far from a boundary.

The SPH approximation of the derivative must also be carefully handled in the presence of a boundary. Firstly, the original integral approximation (12) is adjusted in the same way as

$$\nabla A(\mathbf{r})^\gamma := \frac{\nabla A(\mathbf{r})}{\gamma} = \frac{1}{\gamma} \left[\int_{\partial\Omega} A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|) \mathbf{n} dS + \int_{\Omega} A(\mathbf{r}') \nabla_{\mathbf{r}} W(|\mathbf{r} - \mathbf{r}'|) d\mathbf{r}' \right] \quad (60)$$

Further to the introduction of γ in the equation, the previous assumption made in deriving the original gradient estimate that the surface integral over the boundary $\partial\Omega$ would disappear is no longer accurate. Indeed, near a boundary, the truncated support means that $W(|\mathbf{r} - \mathbf{r}'|)$ is not identically zero along the boundary of integration, and the integral must be included in the final result. This, however, requires careful consideration. The surface integral is over the boundary of the support as opposed to the previous integrals considered which were over the volume of the support. Discretising this integral thus requires the boundary of the support to be well-sampled. Since the integral will only be non-zero along the truncated boundary where the domain boundary intersects, this is equivalent to requiring the domain boundary to be well-sampled i.e. requiring the presence of *boundary* particles.

At this stage, $\mathbf{n}$ is considered as the inward normal to maintain consistency with the literature. Then, applying the standard discrete approximation and the preferred formulation of the gradient operator (as shown in Eq. (20)), a discrete approximation of the gradient of a variable near a boundary is given as

$$\nabla A(\mathbf{r})^\gamma = \frac{\rho_i}{\gamma_i} \sum_j \rho_j V_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla_i W_{ij} \mathbf{e}_{ij} - \frac{\rho_i}{\gamma_i} \sum_w \rho_w \left(\frac{A_i}{\rho_i^2} + \frac{A_w}{\rho_w^2} \right) W_{iw} \mathbf{n}_w dS \quad (61)$$

where $\sum_w$ refers to a sum over local wall particles (specifically *boundary* particles in this case) as described. The subscript w refers to the respective variables defined at said wall particles. The integration differential here is the area of the wall particles, dS , since they are only defined along the boundary.

Following the same methodology, the discrete approximation to the divergence under a semi-analytical framework can be written as

$$\nabla \cdot \mathbf{B}(\mathbf{r})^\gamma = -\frac{1}{\rho_i \gamma_i} \sum_j \rho_j V_j (\mathbf{B}_i - \mathbf{B}_j) \cdot \nabla_i W_{ij} \mathbf{e}_{ij} + \frac{1}{\rho_i \gamma_i} \sum_w \rho_w (\mathbf{B}_i - \mathbf{B}_w) \cdot W_{iw} \mathbf{n}_w dS \quad (62)$$

or

$$\nabla \cdot \mathbf{B}(\mathbf{r})^\gamma = -\frac{1}{\gamma_i} \sum_j V_j (\mathbf{B}_i - \mathbf{B}_j) \cdot \nabla_i W_{ij} \mathbf{e}_{ij} + \frac{1}{\gamma_i} \sum_w (\mathbf{B}_i - \mathbf{B}_w) \cdot W_{iw} \mathbf{n}_w dS \quad (63)$$

It is then also possible to define an appropriate adjustment to the Laplacian. The same methodology as the ‘‘Morris’’ operator (25) is used - a combination of an SPH divergence estimate and a first order finite difference approximation. The SPH estimate is based on the adjusted normalised form as in Eq. (60). This leads to a discretised equation of the form

$$\nabla^2 A(\mathbf{r})^\gamma := \frac{1}{\gamma_i} \sum_j 2 \left(\frac{\mathbf{A}_i - \mathbf{A}_j}{|\mathbf{r}_{ij}|} \right) V_j \nabla_i W_{ij} - \frac{1}{\gamma_i} \sum_w (\nabla \mathbf{A}_i + \nabla \mathbf{A}_w) \cdot \mathbf{n}_w W_{iw} dS \quad (64)$$

Here the finite difference approximation to the gradient has not yet been applied to the summation over wall particles, as it first necessary to approximate $\nabla \mathbf{A} \cdot \mathbf{n}_w$. This is simply the directional derivative of the variable $\mathbf{A}$ perpendicular to the wall. Since the Laplacian is applied to the velocity $\mathbf{u}$, a contextual approximation to this term can be made by assuming the physical behaviour of the velocity field in the vicinity of the wall. Considering the viscous sublayer, velocity is expected to vary linearly with distance to the wall, and it is further assumed that velocity only varies perpendicularly to the wall. This leads to the approximation at first-order $\nabla \mathbf{A}_i \cdot \mathbf{n}_w \approx \nabla \mathbf{A}_w \cdot \mathbf{n}_w \approx \frac{\mathbf{A}_i - \mathbf{A}_w}{|(\mathbf{r}_i - \mathbf{r}_w) \cdot \mathbf{n}_w|}$. This approximation can equivalently be arrived at by considering the laminar shear stress at the wall [53]. The Laplacian approximation is then

$$\nabla^2 \mathbf{A}(\mathbf{r})^\gamma = \frac{1}{\gamma_i} \sum_j 2 \left(\frac{\mathbf{A}_{ij}}{|\mathbf{r}_{ij}|} \right) V_j \nabla_i W_{ij} - \frac{1}{\gamma_i} \sum_w 2 \left(\frac{\mathbf{A}_{iw}}{|\mathbf{r}_{iw} \cdot \mathbf{n}|} \right) W_{iw} dS \quad (65)$$

Here on, the superscript γ for normalised versions of SPH approximations will be dropped, and they will be treated as the standard formulations when employing the semi-analytical boundary method. These adjusted forms of the gradient, divergence, and Laplacian allow for the governing fluid dynamics equations (21) to be rewritten in a semi-analytical framework, and as such essentially define the implementation of the semi-analytical boundary method. In comparison to the original operators presented in Sec. 2.4, it can be seen that full consideration of the boundary integral has led to the appearance of a second term in the derivatives when in the locality a wall boundary. This term, along with the $1/\gamma$ scaling, is what leads to increased forces or densities near a boundary to emulate the boundary effect. In the momentum equation (27), the second term in the gradient (61) generates a pressure gradient pointing towards the wall, whilst in the continuity equation (26), the second term in the divergence (62) generates a density increase when a fluid particle approaches a wall.

Of course, with discretised boundary particles, then the same considerations on boundary particle placement and variable setting must be made, as was for the ghost particles methods (Sec.s 3.2). It is simplified in that the wall itself does not need to be discretised, only the boundary. The most common implementations [53, 105] typically place boundary particles directly on the wall boundary (as is the default expectation), and set the pressure by a local average of fluid particles, as in Eq. (53). These considerations will be discussed in more detail in Chapter 5.

Alternative formulations of these SPH approximations can be found in the literature. One of the earliest formulations of the semi-analytical boundary method was produced by Kulasegaram [86]. More recently, however, Ferrand [53] improved on Kulasegaram's method, and produced a formulation roughly equivalent to that presented here. Notable about Kulasegaram's formulation is that it does not rely on summations over wall particles, and instead evaluates the wall contribution in a single term. Depending on the implementation, this can be preferred. Further, the gradient and divergence operators offered by Kulasegaram are skew-adjoint, being thus compatible with a variational approach and offering ideal conservation properties. However, it was found to be a less accurate formulation in practice than those involving wall particle summations [53]. This is due to the fact that it does not predict zero net force in a uniform pressure field. Furthermore, wall summation methods offer increased resolution of wall variables.

Often, the term in the wall summation $W_{iw} \mathbf{n}_w dS$, as in Eq. (61), is written as $\nabla \gamma_{iw}$. In this context, $\nabla \gamma_{iw}$ refers to the contribution to $\nabla \gamma_i$ from the wall particle at position $\mathbf{r}_w$.

This can be found by considering the gradient of γ in Eq. (58). First passing the ∇ operator (which is here understood as the gradient with respect to $\mathbf{r}_a$) inside the integral using the Leibniz integral rule, and then appropriately flipping the evaluation of the gradient so that both the integral and ∇ are with respect to $\mathbf{r}'$. The divergence theorem can then be applied to produce the relevant equation

$$\nabla\gamma = - \int_{\partial\Omega} W(|\mathbf{r}' - \mathbf{r}|) \mathbf{n} dS \quad (66)$$

Given a wall discretised into particles, a discrete approximation of this is then

$$\nabla\gamma_i = - \sum_w \nabla\gamma_{iw} \quad (67)$$

leading to

$$\nabla\gamma_{iw} \approx W_{iw} \mathbf{n}_w dS \quad (68)$$

From this, the approximation $W_{iw} \mathbf{n}_w dS = \nabla\gamma_{iw}$ can be seen, and there is now a motivation for calculating $\nabla\gamma$ which appears in in the gradient approximation (61).

There also exists an alternative approach to semi-analytical boundaries which is occasionally referenced in the literature [40, 108, 6]. This is produced not by considering the effect of the missing kernel support on the governing equations, but rather by attempting to extend the variables' fields into the boundary region, such that there is effectively no longer a truncation. This can perhaps be thought of as a continuous approximation to a ghost particle approach. Usually, a constant or very simplified field is assumed in the boundary region, such that the only difficulty remaining is to evaluate the integral of the kernel function W or its derivative ∇W over the truncated boundary region. This integration does not generally appear to be addressed in the literature, though the work presented in Chapter 4 could be leveraged to that end. Depending on how integrations or summations are calculated, this approach may also offer the advantage of not requiring any wall particle discretisations.

The semi-analytical boundary method is often seen as a preferred approach to modelling boundaries in SPH due to the fact that it arises naturally from a full consideration of the governing equations. It does not necessitate artificial additions to the method, such as a repulsive force or ghost particle method may. It also restores first-order accuracy in areas where the kernel support is undersampled. In the case of coastal dynamics problems, avoiding the discretisation of wall boundaries can be advantageous due to the potential for highly irregular and transient physical boundaries. Further, as previously discussed, accuracy of solid boundary effects is extremely important and influential on the resulting dynamics, and so there is an impetus to use the most accurate boundary methods available.

The semi-analytical boundary method, however, does forego some of the symmetry and conservation properties of the original governing equations. This is due to the introduction of γ_i which is not symmetric between pairs of particles, and the use of the gradient and divergence operators defined in Eqs (61) and (62) which are no longer skew-adjoint. In particular, loss of antisymmetry in the gradient operator (61) means that momentum is not necessarily exactly conserved near a boundary. However, these issues do disappear far from the boundaries, as the semi-analytical method returns to the original SPH approximation when $\gamma = 1$ and $\nabla\gamma = \mathbf{0}$.

Despite its issues, the semi-analytical boundary method is often perceived to be one of the more accurate boundary modelling approaches [91]. Its lack of widespread use is often owed to the difficulty in accurately calculating the normalisation terms γ and $\nabla\gamma$. In the following chapters, this problem is effectively solved, with the presentation of a fully accurate and general framework for their calculation, as well as further improvements to increase the method's accuracy.

4 An improved semi-analytical boundary method (1) - Calculation of normalisation terms γ and $\nabla\gamma$

The main difficulties in the semi-analytical boundary method have been related to the calculation of γ and its gradient $\nabla\gamma$. Either one, or both of these are required in the adjusted governing equations depending on the formulation. An effective semi-analytical boundary method should include accurate and efficient calculation of these terms, as well as a system of governing equations which involves mathematically consistent SPH operators. Owing to work by Ferrand et al. who presented an effective system of governing equations for the method [53, 54], the focus in recent literature and further development of the method has been the calculation of the γ and $\nabla\gamma$ terms, which is seen as the main source of difficulty in the method. Further, standard calculations of γ and $\nabla\gamma$ are typically only handled near planar boundaries.

In this work, formulations of these normalisation terms near spherical boundaries are also provided. Since most wall boundaries are planar, this has historically been the focus of the semi-analytical method. However, spherical boundaries present themselves when considering solid spherical objects which may behave as boundaries. An example would be mobile DEM bodies within the fluid which represent coastal solids, such as boulders or rocks, and are often modelled as spheres. Calculating γ and $\nabla\gamma$ near spherical boundaries allows the opportunity to model mobile spherical solids with the semi-analytical method - something which has previously not been possible. This section will thus provide a simple introduction to the possibility of accurate semi-analytical fluid-solid coupling for non-planar mobile solids. This does not appear to have been explored before in the literature, other than the publication produced from this work [85].

It is noted that while the original motivation and application of this boundary integral method is the semi-analytical boundary method within SPH, the method described is in fact general and can be applied to any integration of radial functions over truncated, non-standard domains. This is equivalent to the situation in SPH, where the support domain of the radial kernel function is truncated near boundaries, as visualised in Fig. 10.

The difficulty in calculating the normalisation terms owes to the fact that γ is calculated via the integral of a radial function over a truncated, usually polygonal, and potentially concave, domain. Typically these integrals cannot be solved analytically, and so are instead solved either wholly or partly numerically, relying on some discretisation of the boundaries. This often involves very complex and expensive implementations. In this section, a method which effectively solves this issue is presented. The method can solve the γ integrals completely analytically (and partly numerically, if ever required), and is presented

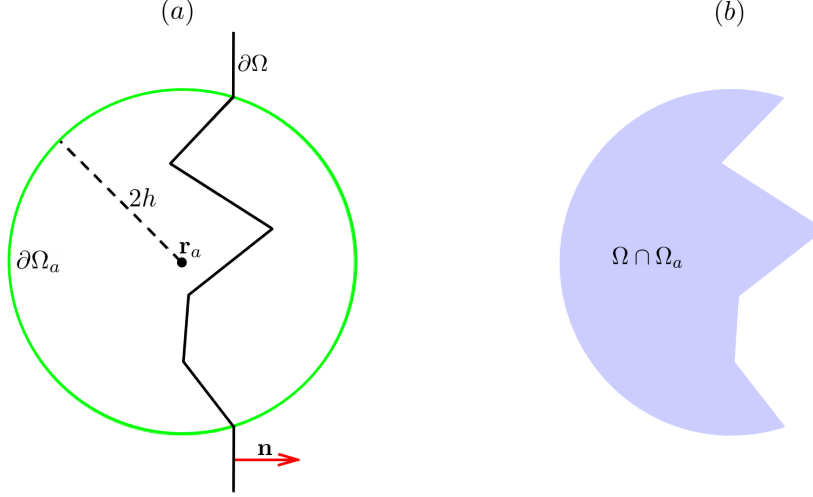


Figure 10: Schematic visualisation of the integration domain $\Omega \cap \Omega_a$ in two dimensions. In (a), a particle positioned at $\mathbf{r}_a$ has circular (spherical) finite support Ω_a with radius R (green). Here the particle is in the vicinity of the boundary of the computational domain Ω (black) with outward normal $\mathbf{n}$ (red). Its effective support is then truncated by $\partial\Omega$, creating a local particle boundary $\partial\Omega_a$. The resulting domain of interest $\Omega \cap \Omega_a$ is visualized in (b).

alongside efficient algorithms for implementation which can be ported to any existing SPH code. Whilst this does not completely solve all issues related to the semi-analytical boundary method, the aim is that it does solve the issue of calculating γ and $\nabla\gamma$ for any polygonal or spherical boundary types. Other issues within the semi-analytical boundary method, such as the choice of governing equations or time integration scheme, will also be discussed later in the section.

Working from Eqs (58) and (66), the aim is to solve the integrals

$$\gamma_a \equiv \int_{\Omega \cap \Omega_a} W(|\mathbf{r}_a - \mathbf{r}'|) d\mathbf{r}' \quad (69)$$

$$\nabla\gamma_a = - \int_{\partial\Omega_a} W(|\mathbf{r}_a - \mathbf{r}'|) \mathbf{n} dS \quad (70)$$

in both two and three dimensions. It is clarified that the integral $\int_{\Omega \cap \Omega_a} d\mathbf{r}'$ will represent an area integral in a two dimensional domain, and a volume integral in a three dimensional domain. The use of the subscript a is equivalent to the previous subscript i and refers only to the particle of interest, but is used here to distinguish from the discrete case where i is used. $\mathbf{n} = \mathbf{n}(\mathbf{r})$ is now defined as the local outward unit normal at the boundary, and dS is a differential element of the boundary of the domain. In two dimensions, the integral $\int_{\partial\Omega_a} dS$ can be thought of as a line integral, and in three dimensions as an area integral.

These integrals should be solvable over a wide range of domain intersections $\Omega \cap \Omega_a$, avoiding domain decomposition and expensive numerical integration where possible.

Early attempts at calculating γ involved creating a polynomial approximation to (69) for simple 2D geometries [86], or limiting the boundary to single straight lines or corners [52]. Ferrand et al. presented the first consistent semi-analytical formulation in 2D [53], which was then extended to 3D by Mayrhofer et al. [105]. However, both of these methods yielded complex and computationally expensive formulations, and particularly in 3D relied on complicated formulae and domain decompositions to calculate the different intersections of a sphere with a triangular planar boundary. These methods are not fully general and do not necessarily handle non-convex geometries correctly.

Very recent developments by Calderon-Sanchez et al. [22] and Chiron et al. [28] have made good progress towards a geometric formulation that handles both 2D and 3D implementations and can handle arbitrary boundaries correctly. This is done in the same manner as in this work, which is by using the divergence theorem to reduce the dimensionality of the problem and produce integrals which can be solved. However, numerical or symbolic integration are used for the final evaluation, producing convoluted and computationally expensive numerical methods. Further, these methods usually restrict the smoothing kernel of choice to the standard Wendland quintic kernel, and rely on an additional meshing algorithm or domain trimming process which is used alongside the method, further complicating matters. Generally there is a limited discussion on the steps of the method for three dimensions or how the work can be implemented.

The present work first uses both the divergence theorem and Stokes' theorem to allow a derivation in both 2D and 3D that leads to a simple formulation and reduces both 2D and 3D problems into a 1D line integral. This line integral can then be solved either numerically, or preferably using a fully exact and analytical closed expression which is also presented. The following section covers the implementation within an SPH framework, and pays close attention to efficiency and generality. Within the computation loop, both γ_a and $\nabla\gamma_a$ can be computed simultaneously and any convex polygonal element can be handled.

The work in the following sections of this Chapter is adapted from three separate papers co-authored with Dr Larry Kostorz⁶ during the course of the PhD. All three of these papers were published [83, 85, 84]. Dr Kostorz was a fellow student during the PhD with whom I collaborated on many different areas of work. For the above papers, Dr Kostorz contributed primarily to the mathematical rigour of the work, while I handled the majority of implementation and result generation. Conceptual work, as well as work on the algorithms and efficiencies of the method was equally split. Standard information such as the mathematical

⁶w.j.kostorz@gmail.com, previously *Department of Earth Science & Engineering, Imperial College London*

theory and implementation algorithms are taken directly from the papers, while the discussion and certain results have been largely expanded upon and are adapted to the context of this work.

4.1 Summary of the method

The following mathematics can be generalised to a description of the method as follows. First, the dimensionality of the problem is reduced, using the divergence theorem to reduce a 3D integral to a 2D one, and then using Stokes' theorem to reduce 2D integral to a 1D one along a boundary. The divergence theorem is applied by rewriting a 3D integral as the integral of a divergence, through the use of an appropriate ancillary function. Similarly, Stokes' theorem is applied by rewriting the new (or existing) 2D integral as the integral of a curl, with the use of another ancillary function. This produces a final integral in only one dimension, in a typically much more tractable problem. Along the way, the problem is also reformulated so that analytical expressions can be leveraged where possible to reduce the reliance on integral evaluations. One-dimensional boundaries are split into segments which can be found using analytical geometrical results, and others which require more involved one-dimensional integrations along straight edges.

Of course, this 1D integral can be solved fairly efficiently numerically, as will be discussed, but we also present here further mathematics to provide an exact and analytical closed-form solution. This is done by providing a recurrence relation solution to the integral. It is noted, that whilst this recurrence relation can be used directly to construct an explicit solution equation, this will be a very long and complex formulae due to the recurrences and functions involved. Rather, we suggest explicitly evaluating the initial recurrence points, and applying the recurrence relation to these numerical values, allowing a final numerical value solution to be calculated with a sequence of simple steps requiring only a few multiplications.

The focus in the following sections will be on the 3D version of the problem, as this covers the scope of the problems of interest. The introductory theory, however, is presented for both two and three dimensions, as the 3D problem collapses to a 2D one after initial application of the divergence theorem. The referenced papers from which this theory follows contain a full discussion of the 2D version of the problem.

4.1.1 Two dimensions

The problem is firstly extended to three dimensions, working in the cylindrical coordinate system $\{\mathbf{e}_\rho, \mathbf{e}_\phi, \mathbf{e}_z\}$ centered at $\mathbf{r}_a$ introducing the $\mathbf{e}_z$ unit vector to Eq. (69). We then apply

Stokes' theorem⁷.

In order to apply Stokes' theorem, it is required to identify a vector field $\mathbf{F}$ which satisfies

$$\nabla \times \mathbf{F} = W(\rho)\mathbf{e}_z \quad (71)$$

with $\rho = \|\mathbf{r} - \mathbf{r}_a\|_{XY}$ (in the XY plane, as in the cylindrical coordinate system i.e. W has only a radial dependence). Considering the symmetry of the problem - radial dependence and pointing only in $\mathbf{e}_z$ - it is possible to suggest a solution of the form

$$\mathbf{F} = f(\rho)\mathbf{e}_\phi \quad (72)$$

The curl in cylindrical coordinates is taken, and then equated to the right-hand side in Eq. (71). For simplicity it is assumed that W is bounded on Ω (which is a valid assumption for any kernel with finite support). It is then found that

$$f(\rho) = \frac{1}{\rho} \int_0^\rho s W(s) ds \quad (73)$$

together with (72) is a solution of (71). By the change of variables $s = \rho\hat{s}$, (where $\hat{s}$ is now a non-dimensional radius), this is written as

$$f(\rho) = \rho \int_0^1 \hat{s} W(\rho\hat{s}) d\hat{s} \quad (74)$$

Furthermore, noting that $\mathbf{e}_\rho = (\mathbf{r} - \mathbf{r}_a) / \rho$, and realising that $\mathbf{e}_\phi$ can be calculated as just a rotation of $\mathbf{e}_\rho$, it can be defined that

$$\begin{aligned} \mathbf{e}_\phi &= \frac{1}{\rho} \mathbf{P}(\mathbf{r} - \mathbf{r}_a), \\ \mathbf{P} &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned} \quad (75)$$

where $\mathbf{P}$ is simply the $\frac{\pi}{2}$ (anti-clockwise) rotation matrix in two dimensions. Together with the solution of form (74) in Eq. (72), Stokes' theorem can be applied to Eq. (69) with $\partial\Omega$ as the boundary bounding curve to find

$$\gamma_a = \int_{\partial\Omega} G_W^{(2)}(\rho) [\mathbf{P}(\mathbf{r} - \mathbf{r}_a)] \cdot d\mathbf{r} = \int_{t_0}^{t_1} G_W^{(2)}(\rho) \dot{\mathbf{r}}^T \mathbf{P}(\mathbf{r} - \mathbf{r}_a) dt, \quad (76a)$$

$$G_W^{(2)}(\rho) = \int_0^1 \hat{s} W(\rho\hat{s}) d\hat{s} \quad (76b)$$

⁷Practically, the two-dimensional simplification of the Stokes' theorem, Green's theorem, is being used. Extending the problem to three dimensions allows us to exploit the radial symmetry of the problem.

with $\mathbf{r} = \mathbf{r}(t)$, $t \in [t_0, t_1]$ being an arbitrary parameterisation of $\partial\Omega$ and $\rho = \rho(t) = |\mathbf{r}_a - \mathbf{r}(t)|$. $\dot{\mathbf{r}}$ is the derivative of $\mathbf{r}$ with respect to the parameterisation variable t along the boundary $\partial\Omega$. It is also noted that the change of integration variable from $\mathbf{r}' \rightarrow \mathbf{r}$ from Eq. (69) is just done for notation consistency and is of no mathematical consequence. The superscript in $G_W^{(2)}$ refers to the number of dimensions of the original integral. For a more comprehensive understanding of the method at this stage, the above steps leading to Eqs (76a) and (76b) are detailed line-by-line in Appendix B. Clarity of the process here should aid the reader's comprehension of the method in a general sense and its extension to three dimensions in the following section.

In practical applications, $G_W^{(2)}(\rho)$ can often be provided explicitly; this allows for evaluating γ_a (originally two-dimensional) via one-dimensional integration. In the case that W is a polynomial, as is almost always the case in SPH, $G_W^{(2)}$ is straightforward to evaluate. In this case

$$W(\rho) = \sum_n a_n \rho^n \Rightarrow G_W^{(2)}(\rho) = \sum_n \frac{a_n}{n+2} \rho^n \quad (77)$$

It should be noted, that when working in two dimensions, $\nabla\gamma_a$ in Eq. (66) is already expressed as a one-dimensional integral along a boundary, and so need not be simplified further, other than parameterising it in t and writing it with the appropriate Jacobian. That is

$$\nabla\gamma_a = - \int_{t_0}^{t_1} W(\rho) (\dot{\mathbf{r}}^T \mathbf{P} \mathbf{n}) \mathbf{n} dt \quad (78)$$

where in the above $\mathbf{n} = \mathbf{n}(\mathbf{r}(t))$.

4.1.2 Three dimensions

The problem in three dimensions is tackled following a similar methodology to that in two dimensions. Firstly, evaluating γ_a addressed. In the 3D form of the problem we want to apply the divergence theorem, and therefore look for a field $\mathbf{F}$ satisfying

$$\nabla \cdot \mathbf{F} = W(r) \quad (79)$$

where $r = \|\mathbf{r} - \mathbf{r}_a\|$ and $\{\mathbf{e}_r, \mathbf{e}_\phi, \mathbf{e}_\theta\}$ being the spherical coordinate system centered at $\mathbf{r}_a$. We again use symmetry of the problem to suggest a solution of the form

$$\mathbf{F} = f(r) \mathbf{e}_r \quad (80)$$

This is then used in Eq. (79) expressed in spherical coordinates. By taking the divergence in spherical coordinates, assuming boundedness of W on Ω , and an appropriate change of variables, it can be verified that

$$f(r) = \frac{1}{r^2} \int_0^r s^2 W(s) \, ds = r \int_0^1 \hat{s}^2 W(r\hat{s}) \, d\hat{s} \quad (81)$$

together with (80) is a solution of (79). We then use

$$\mathbf{e}_r = \frac{1}{r}(\mathbf{r} - \mathbf{r}_a) \quad (82)$$

in Eq. (80).

By combining these equations and applying the divergence theorem to Eq. (69) it follows that

$$\gamma_a = \int_{\partial\Omega} G_W^{(3)}(r) (\mathbf{r} - \mathbf{r}_a)^T \mathbf{n} \, dS \quad (83a)$$

$$G_W^{(3)}(r) = \int_0^1 \hat{s}^2 W(r\hat{s}) \, d\hat{s} \quad (83b)$$

where $\mathbf{n}$ is the outward unit normal to $\partial\Omega$, of which dS is a differential element. In practice $G_W^{(3)}$ can often be provided explicitly; an explicit approximation should be used otherwise. Similarly to the two-dimensional case, $G_W^{(3)}(r)$ can be evaluated easily for polynomial W as follows

$$W(r) = \sum_n a_n r^n \Rightarrow G_W^{(3)}(r) = \sum_n \frac{a_n}{n+3} r^n \quad (84)$$

Next, let us assume that $\partial\Omega$ is piecewise-planar. Consider an arbitrary planar portion (e.g. a triangle, common for unstructured meshing of boundaries) of $\partial\Omega$, written $\partial\Omega_j$. It can be clarified that

$$\begin{aligned} \gamma_a &= \sum_j \gamma_j \\ \nabla \gamma_a &= \sum_j \nabla \gamma_j \end{aligned} \quad (85)$$

where j then runs over all boundary elements, and γ_j denotes the contribution to γ_a from the j -th boundary element and equivalently for $\nabla \gamma_j$. This is to say that each boundary element $\partial\Omega$ intersection can be broken down into its constituent portions $\partial\Omega_j$.

We denote the plane of $\partial\Omega_j$ as P with a constant unit normal $\mathbf{n}$ (the outward normal for $\partial\Omega_j$) and the projection of $\mathbf{r}_a$ onto P as $\mathbf{r}_P$. In this case for any $\mathbf{r} \in P$ we have

$$(\mathbf{r} - \mathbf{r}_a)^T \mathbf{n} = (\mathbf{r}_P - \mathbf{r}_a)^T \mathbf{n} := \lambda \quad (86)$$

for some constant λ , satisfying $\lambda^2 = \|\mathbf{r}_P - \mathbf{r}_a\|^2$, which is the signed perpendicular distance to the plane P from $\mathbf{r}_a$. Additionally, for any $\mathbf{r} \in P$ it is true that

$$\|\mathbf{r} - \mathbf{r}_a\|^2 = \|\mathbf{r} - \mathbf{r}_P\|^2 + \lambda^2 \quad (87)$$

and, by defining $\rho = \|\mathbf{r} - \mathbf{r}_P\|$, Eq. (83) can be rewritten as

$$\gamma_j = \lambda \int_{\partial\Omega_j} G_W^{(3)} \left(\sqrt{\rho^2 + \lambda^2} \right) dS \quad (88)$$

Since $G_W^{(3)} \left(\sqrt{\rho^2 + \lambda^2} \right)$ is radially symmetric around $\mathbf{r}_P$ in the plane P , and we now have a two-dimensional surface integral, it is possible to reapply the methodology from the previous section in order to further reduce the dimensionality of the problem.

The $\frac{\pi}{2}$ rotation matrix $\mathbf{P}$ acting on $\mathbf{r} - \mathbf{r}_a$ in Eq. (76a) must be replaced by a $\frac{\pi}{2}$ rotation in the plane P acting on $\mathbf{r} - \mathbf{r}_P$, namely

$$\mathbf{P}(\mathbf{r} - \mathbf{r}_a) \rightarrow \mathbf{n} \times (\mathbf{r} - \mathbf{r}_P) \quad (89)$$

The analogue to Eq. (76b) can be defined as

$$H_W^{(3)}(\rho; {}^\dagger\lambda) = \int_0^1 \hat{s} G_W^{(3)} \left(\sqrt{\rho^2 \hat{s}^2 + \lambda^2} \right) d\hat{s} \quad (90)$$

In order to repeat the dimensionality reduction process using Stokes' theorem, this integral must be transformed to an appropriate form as in Eq. (76a). This is simply done by updating the dummy variable as $\hat{s} = \hat{s}^2$ to give

$$H_W^{(3)}(\rho; \lambda) = \frac{1}{2} \int_0^1 G_W^{(3)} \left(\sqrt{\rho^2 \hat{s} + \lambda^2} \right) d\hat{s} \quad (91)$$

The analogue to Eq. (76a) can now be applied resulting in

$$\gamma_j = \lambda \int_{\partial\Omega_j} G_W^{(3)} \left(\sqrt{\rho^2 + \lambda^2} \right) dS = \lambda \int_{C_j} H_W^{(3)}(\rho(t); \lambda) \mathbf{n}^T ((\mathbf{r} - \mathbf{r}_P) \times \dot{\mathbf{r}}) dt \quad (92)$$

with C_j being the contour of $\partial\Omega_j$ oriented anti-clockwise with respect to $\mathbf{n}$ and $\mathbf{r} = \mathbf{r}(t)$ being a parameterisation of C_j . The identity $\mathbf{a}^T(\mathbf{b} \times \mathbf{c}) = \mathbf{b}^T(\mathbf{c} \times \mathbf{a})$ was also used.

[†]The use of the semi-colon ; to separate the arguments of $H_W^{(3)}$ is intended to signify the difference of λ as a fixed parameter and ρ as a variable when this function is part of an integrand, as $\rho = \rho(t)$ during integration. In practice this does not affect the implementation or understanding of the function - it is just for clarity.

$H_W^{(3)}$ can also be provided explicitly for any polynomial W . Explicit integration yields

$$W(r) = \sum_n a_n r^n \Rightarrow H_W^{(3)}(\rho; \lambda) = \sum_n \frac{a_n}{(n+2)(n+3)} |\lambda|^n \frac{(1+\kappa)^{\frac{n}{2}+1} - 1}{\kappa}, \quad \kappa = \frac{\rho^2}{\lambda^2} \quad (93)$$

together with the limiting cases

$$\begin{aligned} H_W^{(3)}(0; \lambda) &= \sum_n \frac{a_n}{2(n+3)} |\lambda|^n \\ H_W^{(3)}(\rho; 0) &= \sum_n \frac{a_n}{(n+2)(n+3)} \rho^n \end{aligned} \quad (94)$$

Finally, with analogous notation and derivation (comparing to Eqs (91) and (92)), for a planar $\partial\Omega_j$, we have

$$\nabla \gamma_j = -\mathbf{n} \int_{\partial\Omega_j} W \left(\sqrt{\rho^2 + \lambda^2} \right) dS = -\mathbf{n} \int_{C_j} H_W^{(2)}(\rho(t); \lambda) \mathbf{n}^T ((\mathbf{r} - \mathbf{r}_P) \times \dot{\mathbf{r}}) dt \quad (95a)$$

$$H_W^{(2)}(\rho; \lambda) = \frac{1}{2} \int_0^1 W \left(\sqrt{\rho^2 \hat{s} + \lambda^2} \right) d\hat{s} \quad (95b)$$

$H_W^{(2)}(\rho; \lambda)$ for polynomial W is similar to $H_W^{(3)}(\rho; \lambda)$ in Eqs (93) and (94)

$$\begin{aligned} H_W^{(2)}(\rho; \lambda) &= \sum_n \frac{a_n}{(n+2)} |\lambda|^n \frac{(1+\kappa)^{\frac{n}{2}+1} - 1}{\kappa}, \quad \kappa = \frac{\rho^2}{\lambda^2} \\ H_W^{(2)}(0; \lambda) &= \sum_n \frac{a_n}{2} |\lambda|^n \\ H_W^{(2)}(\rho; 0) &= \sum_n \frac{a_n}{(n+2)} \rho^n \end{aligned} \quad (96)$$

4.2 Simple planar Boundaries in 3D

In the previous sections, general formulae were provided for converting surface and volume integrals over domains with piecewise-planar boundaries into (sums of) line integrals. Whilst the formulation in Sec.s 4.1.1 and 4.1.2 can be applied to any arbitrary boundary element, here we consider two types of contours C_j which are of particular interest in this work. The first one is a straight line which enables integrating over domains bounded by polygonal (e.g. triangular) meshes - this being the typical way of representing geometries in numerical models. Second, we are interested in (a part of) C_j being an arc of a circle in the plane P of the boundary element $\partial\Omega_j$. This shape occurs naturally in the SPH method due to the intersections of a spherical kernel function with a polygonal planar boundary (an example of this can be seen in Fig. 11). The appropriate parameterisation will be provided, and in doing so, simplify the formulation further for implementation in an SPH method.

4.2.1 Piecewise-linear segment

For a boundary element $\partial\Omega_j$ with contour C_j and constant unit normal $\mathbf{n}$, consider a straight line segment $C_j^{(i)}$ of C_j , in the plane P and spanned by $\mathbf{x}_{1,2}$. The unit direction along the segment is then

$$\hat{\mathbf{e}} = \frac{\mathbf{x}_2 - \mathbf{x}_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|} \quad (97)$$

The orthogonal projection $\mathbf{r}_l$ of $\mathbf{r}_P$ onto the line l passing through $\mathbf{x}_{1,2}$ is then given by

$$\mathbf{r}_l = \mathbf{x}_1 + (\hat{\mathbf{e}}^T(\mathbf{r}_P - \mathbf{x}_1)) \hat{\mathbf{e}} \quad (98)$$

The following arc length parameterisation of $C_j^{(i)}$ centered at $\mathbf{r}_l$ is used

$$\mathbf{r}(t) = \mathbf{r}_l + t \hat{\mathbf{e}}, \quad t \in [t_1, t_2] \quad (99)$$

where

$$t_i = \hat{\mathbf{e}}^T(\mathbf{x}_i - \mathbf{r}_l) \quad (100)$$

Nothing the orthogonal projection definitions of $\mathbf{r}_P$ from $\mathbf{r}_a$, and $\mathbf{r}_l$ from $\mathbf{r}_P$, it can be noted that $\mathbf{r}_l$ lies in the same perpendicular plane to l as $\mathbf{r}_a$ (Fig. 11), and thus

$$t_i = \hat{\mathbf{e}}^T(\mathbf{x}_i - \mathbf{r}_a) \quad (101)$$

Substituting the above parameterisation (99) into Eq. (92), the contribution to γ_j along $C_j^{(i)}$, denoted $\gamma_j^{(i)}$, can be simplified to

$$\gamma_j^{(i)} = \mu \lambda \int_{t_1}^{t_2} H_W^{(3)}(\rho(t); \lambda) dt \quad (102)$$

with

$$\mu := \mathbf{n}^T((\mathbf{r}_l - \mathbf{r}_P) \times \hat{\mathbf{e}}) \quad (103)$$

being the signed distance between $\mathbf{r}_P$ and $\mathbf{r}_l$ (Fig. 11). Notably, it can be seen that the term $\mathbf{n}^T((\mathbf{r} - \mathbf{r}_P) \times \dot{\mathbf{r}})$ has lost its dependence on t , owing to the parameterisation, and as such can be taken outside of the integral. Noting that $\mathbf{x}_1$ and $\mathbf{r}_l$, and $\mathbf{r}_a$ and $\mathbf{r}_P$ lie in the same respective parallel planes, it can be written that

$$\mu = \mathbf{n}^T((\mathbf{x}_1 - \mathbf{r}_a) \times \hat{\mathbf{e}}) \quad (104)$$

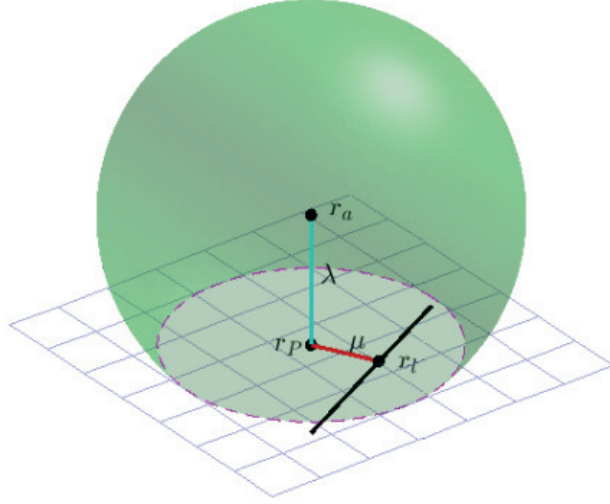


Figure 11: A spherical isosurface of a radial function (green) in the vicinity of a straight edge (black) passing through $\mathbf{x}_{1,2}$ (not shown) in a given plane P (grey mesh). The signed distance λ between the centre of the function $\mathbf{r}_a$ and its projection $\mathbf{r}_P$ onto the plane is plotted in light blue. The signed distance μ between $\mathbf{r}_P$ and its projection $\mathbf{r}_l$ onto the line l containing the edge is plotted in red. For $\mathbf{x}$ along the edge we have $\|\mathbf{x} - \mathbf{r}_a\|^2 = \lambda^2 + \rho^2 = \lambda^2 + \mu^2 + \|\mathbf{x} - \mathbf{r}_l\|^2 = \lambda^2 + \mu^2 + t^2$. In our notation t is the appropriately oriented arc length parameterisation of l with $t = 0$ corresponding to $\mathbf{r}_l$. The parameterisation limits $t_{1,2}$ will depend on $\mathbf{x}_{1,2}$ following Eq. (101), though as mentioned, these are not shown here. They would generally be understood as the parameterisation of either the points where the edge intersects the sphere, or in the case when it does not, the respective endpoints of the edge.

From Pythagoras' theorem, ρ , which defined the distance from a point in the plane P to $\mathbf{r}_P$, can be written

$$\rho(t)^2 = \mu^2 + t^2 \quad (105)$$

In this formulation, both (101) and (104) can be evaluated without constructing $\mathbf{r}_P$ and $\mathbf{r}_l$ explicitly, but with employing just the centre of the radial function and the end points of the edge; such treatment makes implementations more efficient.

The contribution to $\nabla \gamma_j$ along $C_j^{(i)}$ (from Eq. (95a)), denoted $\nabla \gamma_j^{(i)}$, can be written as

$$\nabla \gamma_j^{(i)} = -\mu \mathbf{n} \int_{t_1}^{t_2} H_W^{(2)}(\rho(t); \lambda) dt \quad (106)$$

4.2.2 Circular arc segment

Next, consider a segment k of the contour C_j which is an arc of a circle in the plane P with constant radius $\rho = \rho_\perp$ centered at $\mathbf{r}_P$. The angle spanned by the arc is denoted as ψ , and thus its length is $\psi \rho_\perp$. This circular arc can be parameterised according to its arc length s with $t \in [0, 1]$, as $s = \psi \rho_\perp = \int_0^1 \|\dot{\mathbf{r}}\| dt$. It is therefore followed with constant $\|\dot{\mathbf{r}}\| = \psi \rho_\perp$.

Noting that $\|\mathbf{r} - \mathbf{r}_P\| = \rho_\perp$, and that $((\mathbf{r} - \mathbf{r}_P) \times \dot{\mathbf{r}})$ points in the same direction as $\mathbf{n}$, it can be found

$$\mathbf{n}^T ((\mathbf{r} - \mathbf{r}_P) \times \dot{\mathbf{r}}) = \pm \psi \rho_\perp^2 \quad (107)$$

with the sign corresponding to the direction in which the arc is followed around $\mathbf{n}$. As the circular arc lies in the plane P , λ (the signed perpendicular distance to P from $\mathbf{r}_a$) is constant, along with $\rho = \rho_\perp$. This means that both $H_W^{(2,3)}(\rho; \lambda)$ are constant along such an arc. Therefore the contributions to γ_a (92) and $\nabla \gamma_a$ (95a) are

$$\begin{aligned} \gamma_j^{(k)} &= \pm \lambda \psi \rho_\perp^2 H_W^{(3)}(\rho_\perp; \lambda) \\ \nabla \gamma_j^{(k)} &= \mp \psi \rho_\perp^2 \mathbf{n} H_W^{(2)}(\rho_\perp; \lambda) \end{aligned} \quad (108)$$

and hence the line integral can be evaluated directly with no explicit integration involved, as long as the direction (orientation) information is provided.

4.3 Spherical boundaries in 3D

Consider a spherical boundary represented by a ball Ω' with radius R_b be distance αR_b away from the centre of the SPH particle. The ratio of the radii $\frac{2h}{R_b}$ of the SPH particle and Ω' is written as β , so that the radius of the SPH particle is $2h = \beta R_b$. For now it is assumed that $\beta \geq 1$, i.e. the solid boundary is always smaller than or equal to SPH particle support. The derivation for the case $\beta < 1$ is fully analogous and briefly discussed in the last subsection. Which of these two cases are more relevant depends on the implementation required, but ultimately both can be treated the same, and so the focus here is given to the case of boundaries smaller than the SPH particle radius.

An example problem situation is presented in Fig. 12, where the support of an SPH fluid particle may be overlapping with both circular and planar boundaries, leading to an incomplete support Ω for the kernel function. The circular boundary intersection will be now addressed; the planar boundary intersection has already been discussed the previous section.

The contributions of the spherical boundary to γ_a and $\nabla \gamma_a$ are defined as

$$\gamma^- = - \int_{\Omega \cap \Omega'} W(|\mathbf{r}' - \mathbf{r}_a|) \mathbf{r}' \quad (109a)$$

$$\nabla \gamma^- = \int_{\partial(\Omega \cap \Omega')} W(|\mathbf{r}' - \mathbf{r}_a|) \mathbf{n} dS \quad (109b)$$

where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the local outward normal to the boundary surface $\partial(\Omega \cap \Omega')$. The negative

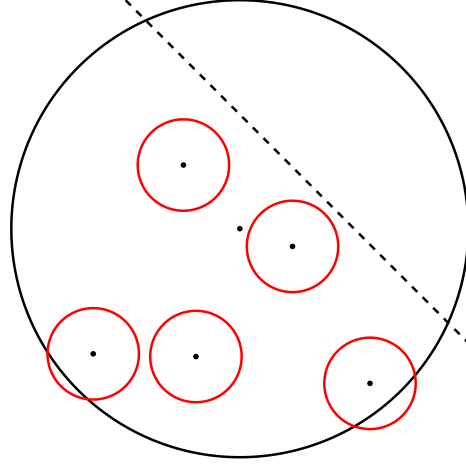


Figure 12: A visual outline of an example problem case. An SPH fluid particle (black, solid) may be intersecting with both mobile solids within the flow (red) and wall boundaries (black, dashed). Both of these define separate regions where the kernel support is incomplete. Here the orientation of the straight boundary is such that the centre of the SPH particle is drawn on the inside of the fluid domain.

signs have been introduced as the spherical boundary intersection necessarily subtract and add to the initialisations of γ_a and $\nabla\gamma_a$ respectively. Consequently, for an SPH particle

$$\begin{aligned}\gamma_a &= \gamma_{\text{bnd}} + \sum_j \gamma_j^- \\ \nabla\gamma_a &= \nabla\gamma_{\text{bnd}} + \sum_j \nabla\gamma_j^-\end{aligned}\tag{110}$$

where γ_{bnd} and $\nabla\gamma_{\text{bnd}}$ are evaluated in the absence of spherical boundaries but taking into account planar boundaries, as discussed in Sec. 4.2. In the above the index j runs over all spherical boundaries with non-trivial intersections with the particle support. Note that formulation 110 is valid only if the spherical boundaries do not overlap. Whilst this was a standard and trivial consideration when handling planar boundaries, here this condition should be explicitly stated, as this method may be employed for mobile boundaries and/or solid particles which may not have such explicit non-overlap criteria.

The distance between the centres of the spherical boundary and SPH particles is written as αR for some $\alpha \geq 0$. The variables α and β can be used to encode the relative position of the center of the spherical boundary to the center and boundary of the SPH particle. There are three cases, also visualized in Fig. 13, to be considered in the remainder of this section:

1. $\alpha = 0$, the boundary and SPH particle are concentric

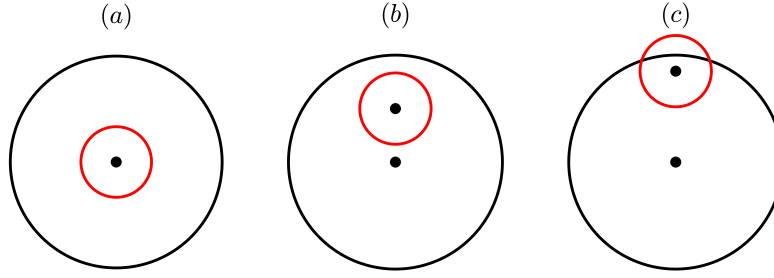


Figure 13: Three cases of non-trivial contributions of a spherical boundary (red) to the normalisation factor, γ_a , of the SPH particle (black) and its gradient, $\nabla\gamma_a$ (schematic side view).

2. $0 < \alpha \leq \beta - 1$, the boundary is fully contained within the SPH particle support; the two are not concentric
3. $\beta - 1 < \alpha < \beta + 1$, the boundary and SPH particle support have a non-trivial intersection

Additionally, the trivial case $\beta + 1 \leq \alpha$ corresponds to the boundary being completely outside of the support of the SPH particle and results in $\gamma^- = 0$, $\nabla\gamma^- = \mathbf{0}$.

4.3.1 Case (a): $\alpha = 0$

In practice this case is rarely encountered, as repulsive mechanisms due to the solid boundary implementation tend to prevent excessive overlap between SPH particles and solid boundaries. In the case of $\beta < 1$ - boundaries larger than SPH particles - this may even be explicitly disallowed as it would imply the SPH particle has entered into the solid boundary and “leaked” from the system (Sec. 3.2.1). However, in the case of $\beta \geq 1$ - boundaries smaller than SPH particles - it could be possible that repulsion mechanisms are not large enough and so an SPH particle may lie concentrically on a solid boundary. This is not necessarily unintuitive, as there are many examples in fluid-solid coupled problems where small suspended solids are completely submerged within the fluid domain.

By the symmetry of the problem it immediately follows that $\nabla\gamma^- = \mathbf{0}$. The evaluation of γ^- can be performed by expressing (83) in spherical coordinates (r, ϕ, θ) centered at the common centre of Ω, Ω' with

$$\begin{aligned} r &= R_b = \text{const.} \\ (\mathbf{r} - \mathbf{r}_a)^T \mathbf{n} &= R_b \end{aligned} \tag{111}$$

resulting in

$$\gamma^- = -4\pi R^3 G_W^{(3)}(R_b) \quad (112)$$

4.3.2 Case (b): $0 < \alpha \leq \beta - 1$

Here $\Omega \cap \Omega' = \Omega'$ since $\Omega' \subset \Omega$. The problem is modelled by centering the coordinate system at the centre of Ω' , and reorienting it so that the centre of the SPH particle is

$$\mathbf{r}_a = R_b[0, 0, -\alpha]^T \quad (113)$$

in Cartesian coordinates. Along $\partial\Omega'$, the defining geometry can be rewritten as

$$\begin{aligned} \mathbf{n} &= [\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta]^T \\ \mathbf{r} &= R_b \mathbf{n} \\ \mathbf{r} - \mathbf{r}_a &= R_b[\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta + \alpha]^T \\ \phi &\in (0, 2\pi), \theta \in (0, \pi) \end{aligned} \quad (114)$$

which after substituting into (83) and using the appropriate jacobian for a three dimensional surface integral, yield

$$\gamma^- = -2\pi R_b^3 \int_0^\pi \sin \theta (1 + \alpha \cos \theta) G_W^{(3)}(R_b \sqrt{1 + 2\alpha \cos \theta + \alpha^2}) d\theta \quad (115)$$

Next, by substituting

$$t^2 = 1 + 2\alpha \cos \theta + \alpha^2 \quad (116)$$

such that $2t dt = -2\alpha \sin \theta d\theta$, Eq. (115) can be rewritten as

$$\gamma^- = -\frac{\pi R_b^3}{\alpha} \int_{|\alpha-1|}^{\alpha+1} (t^3 - (\alpha^2 - 1)t) G_W^{(3)}(R_b t) dt \quad (117)$$

which generally can be evaluated via an arbitrary one-dimensional quadrature. However, for polynomial $W(r)$, as is very often the case in SPH, this can be evaluated directly as follows

$$\begin{aligned} W(r) &= \sum a_n r^n \\ \Rightarrow \gamma^- &= -\frac{\pi}{\alpha} \sum \frac{a_n R_b^{n+3}}{n+3} \left(\frac{(\alpha+1)^{n+4} - |\alpha-1|^{n+4}}{n+4} - (\alpha^2 - 1) \frac{(\alpha+1)^{n+2} - |\alpha-1|^{n+2}}{n+2} \right) \end{aligned} \quad (118)$$

By the analogous derivation starting from Eq. (66) and again substituting (116), it can be verified that

$$\nabla \gamma^- = \hat{\mathbf{e}} \frac{\pi R_b^2}{\alpha^2} \int_{|\alpha-1|}^{\alpha+1} (t^3 - (\alpha^2 + 1)t) W(R_b t) dt \quad (119)$$

where $\hat{\mathbf{e}}$ is the unit direction between the centres of the SPH particle and spherical boundary. Note that the direction of the gradient of γ^- has to be parallel to $\hat{\mathbf{e}}$ by symmetry of the problem, which is confirmed by the derivation leading to Eq. (119).

For polynomial $W(r)$, the integral can again be evaluated directly to

$$\begin{aligned} W(r) &= \sum a_n r^n \\ \Rightarrow \nabla \gamma^- &= \hat{\mathbf{e}} \frac{\pi}{\alpha^2} \sum a_n R_b^{n+2} \left(\frac{(\alpha+1)^{n+4} - |\alpha-1|^{n+4}}{n+4} - (\alpha^2+1) \frac{(\alpha+1)^{n+2} - |\alpha-1|^{n+2}}{n+2} \right) \end{aligned} \quad (120)$$

4.3.3 Case (c): $\beta - 1 < \alpha < \beta + 1$

In this case the surface $\partial(\Omega \cap \Omega')$ is first split into two disjoint parts

$$\begin{aligned} \partial(\Omega \cap \Omega') &= S_1 \sqcup S_2 \\ S_1 &= \partial\Omega' \cap \Omega \\ S_2 &= \partial\Omega \cap \Omega' \end{aligned} \quad (121)$$

S_1 here represents the intersection of the spherical solid boundary within the SPH support, and S_2 represents intersection of the SPH support boundary within the spherical solid boundary.

The integral over S_1 is performed as in the previous section, by centering the coordinate system at the centre of Ω' . However, the lower limit of θ in Eq. (115) ($\theta = 0$) has to be changed to some $\theta_0 > 0$, as visualized in Fig. 14. Consequently, the upper limit (due to the additional minus sign which flips the limits) $\alpha + 1$ in Eqs (117), (119) and results (118), (120), has to be changed to

$$\alpha + 1 \rightarrow \sqrt{1 + 2\alpha \cos \theta_0 + \alpha^2} = \beta \quad (122)$$

with the latter equality found from the geometry of the problem (Fig. 14).

The integral over S_2 is performed by centering the spherical coordinate system at the centre of the SPH particle. With such convention the range of the azimuth θ is $(0, \theta_1)$ (Fig. 14). The necessary angles $\cos \theta_0$, $\cos \theta_1$ can be determined from standard trigonometric relations.

It then follows that the integral (83a) over S_2 reduces to

$$\int_{S_2} G_W^{(3)}(r) (\mathbf{r} - \mathbf{r}_a)^T \mathbf{n} dS = 2\pi(\beta R_b)^3 G_W^{(3)}(\beta R_b) \int_0^{\theta_1} \sin \theta d\theta = \frac{1 - (\alpha - \beta)^2}{4\alpha\beta} \quad (123)$$

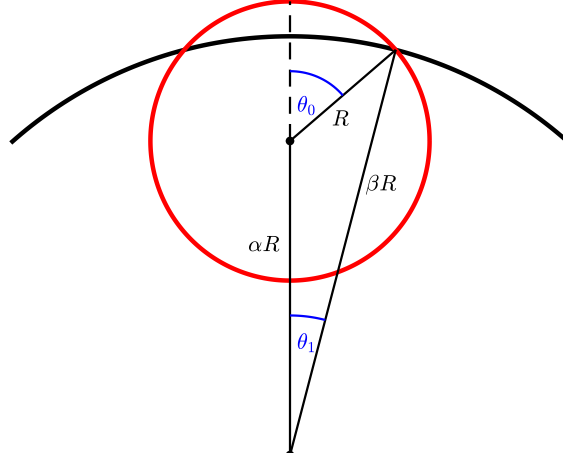


Figure 14: Geometry of the case $\beta - 1 < \alpha < \beta + 1$. The integral over the boundary surface is split into two parts and expressed in spherical coordinates with different origins. The appropriate azimuth bounds $\theta_{0,1}$ are marked in blue (schematic side view).

since

$$1 - \cos \theta_1 = \frac{1 - (\alpha - \beta)^2}{2\alpha\beta}$$

$$G_W^{(3)}(r = R_b) = \frac{1}{4\pi(R_b)^3} \quad (124)$$

with the latter being true for an arbitrary W normalized as in (7). Finally, the contribution to $\nabla\gamma^-$ (based on Eq. (66)) from S_2 is identically zero due to the finite support of the kernel which necessitates $W(r \geq \beta R_b) = 0.0$. Combining the contributions from $S_{1,2}$ yields

$$\gamma^- = \frac{(\alpha - \beta)^2 - 1}{4\alpha\beta} - \frac{\pi R^3}{\alpha} \int_{|\alpha-1|}^{\beta} (t^3 - (\alpha^2 - 1)t) G_W^{(3)}(Rt) dt$$

$$\nabla\gamma^- = \hat{\mathbf{e}} \frac{\pi R^2}{\alpha^2} \int_{|\alpha-1|}^{\beta} (t^3 - (\alpha^2 + 1)t) W(Rt) dt \quad (125)$$

and for polynomial W the above can be given analytically via

$$W(r) = \sum a_n r^n \Rightarrow \begin{cases} \gamma^- = \frac{(\alpha-\beta)^2-1}{4\alpha\beta} - \frac{\pi}{\alpha} \sum \frac{a_n R^{n+3}}{n+3} \left(\frac{\beta^{n+4}-|\alpha-1|^{n+4}}{n+4} - (\alpha^2-1) \frac{\beta^{n+2}-|\alpha-1|^{n+2}}{n+2} \right) \\ \nabla\gamma^- = \hat{\mathbf{e}} \frac{\pi}{\alpha^2} \sum a_n R^{n+2} \left(\frac{\beta^{n+4}-|\alpha-1|^{n+4}}{n+4} - (\alpha^2+1) \frac{\beta^{n+2}-|\alpha-1|^{n+2}}{n+2} \right) \end{cases} \quad (126)$$

4.3.4 Case $\beta < 1$ - Boundaries larger than SPH particles

So far it has been assumed that the solid spherical boundaries were at most as large as SPH particles. The other case, where the solid boundaries are larger, is fully analogous and

simpler. It is clear that for spherical boundaries larger than SPH particles, only the case $1 - \beta < \alpha < 1 + \beta$ is non-trivial; $\alpha \leq 1 - \beta$ indicates that the fluid particle is fully within the solid (and therefore outside the fluid domain and considered “leaked”) while $\alpha \geq 1 + \beta$ results in no contribution to the normalization factor. This is to say, there is either a partial overlap, or none at all.

The case $\beta < 1$ can thus be fully handled by the work in Sec. 4.3.3, Eqs (125), (126).

4.4 Closed-form solution of integrals

Considering both planar and spherical boundaries, equations have now been presented in Sec.s 4.2 and 4.3 for γ_a and $\nabla\gamma_a$. The circular arc planar boundary and spherical boundary interactions are all fully analytical and closed-form for polynomial W , however the straight line boundary interactions in Eqs (102) and (106) remain as 1D integrals to be solved.

These integrals can be solved straightforwardly utilising a numerical integration method. Previous work showed an implementation of this using a custom adaptive 1D numerical integration [83]. However, in a C++ implementation this can also be done using the standard *Boost* libraries [18]. The *Boost* library contains different types of numerical quadrature within its *Math Toolkit*, which are able to integrate arbitrary functions with a given number of evaluation points, N . The *Gauss-Legendre* quadrature implementation was found to work effectively for the given integrals (102) (106), as it is exact for polynomials of order less than $2N$ (as all Gaussian quadratures are), the method is very efficient for integrating polynomials over the domain $[-1, 1]$, and no error estimate is required in the result. Particularly, since the integrands $H_W^{(3)}$ and $H_W^{(2)}$ remain 5th order polynomials, Gauss-Legendre quadrature can produce exact results with only 3 sample points. The standard Gauss-Legendre *Boost* implementation includes pre-computed weights/abscissa for only as few as 7 points, however documentation is available which allows pre-computing the values for the required number of points [19].

The planar boundary straight-line integrals are generally not solved analytically. The reason is a non-trivial dependence of the integrands on ρ and λ , which together with ρ itself being a non-linear function of the natural parameterisation makes it difficult to generate closed-form results. The closest previous result in the literature is the work of Chiron et al. [28], who produced a closed-form expression for their evaluation of γ_a . However, this is in an extremely long and convoluted formula, is only valid for the Quintic Wendland kernel, and does not address the evaluation of $\nabla\gamma_a$.

Here a procedure is provided for generating closed-form anti-derivatives of $H_W^{(2,3)}$ and a method for evaluating them without constructing the analytic expressions explicitly.

Substituting $H_W^{(3)}$ (93) into Eq. (102) and considering only the non-limiting cases $\rho, \lambda \neq 0$, the equation to solve is

$$\gamma_j^{(i)} = \mu\lambda \sum \frac{a_n}{(n+2)(n+3)} |\lambda|^{n+1} \int_{t_1/|\lambda|}^{t_2/|\lambda|} \frac{\left(1 + \frac{\mu^2}{\lambda^2} + s^2\right)^{\frac{n}{2}+1} - 1}{\frac{\mu^2}{\lambda^2} + s^2} ds \quad (127)$$

where the change of variable $t = |\lambda|s$ has been used and of course n remains the order of the polynomial kernel. For $\zeta > 0$, the integral is defined as

$$I_\zeta(x; \phi^2) = \int^x \frac{(1 + \phi^2 + s^2)^\zeta - 1}{\phi^2 + s^2} ds \quad (128)$$

so that with the above notation (and analogously for $\nabla\gamma_j^{(i)}$ from Eqs (96) and (106))

$$\begin{aligned} \gamma_j^{(i)} &= \mu\lambda \sum \frac{a_n}{(n+2)(n+3)} |\lambda|^{n+1} \left[I_{\frac{n}{2}+1} \left(\frac{t_2}{|\lambda|}; \frac{\mu^2}{\lambda^2} \right) - I_{\frac{n}{2}+1} \left(\frac{t_1}{|\lambda|}; \frac{\mu^2}{\lambda^2} \right) \right] \\ \nabla\gamma_j^{(i)} &= -\mathbf{n} \mu \sum \frac{a_n}{n+2} |\lambda|^{n+1} \left[I_{\frac{n}{2}+1} \left(\frac{t_2}{|\lambda|}; \frac{\mu^2}{\lambda^2} \right) - I_{\frac{n}{2}+1} \left(\frac{t_1}{|\lambda|}; \frac{\mu^2}{\lambda^2} \right) \right] \end{aligned} \quad (129)$$

The problem is therefore solved if I_ζ can be evaluated for half-integer $\zeta \geq 1$. An ancillary integral is defined

$$J_\zeta(x; \phi^2) := I_{\zeta+1}(x; \phi^2) - I_\zeta(x; \phi^2) = \int^x (1 + \phi^2 + s^2)^\zeta ds \quad (130)$$

and via integration by parts

$$J_\zeta(x; \phi^2) = x(1 + \phi^2 + x^2)^\zeta - 2\zeta \int^x s^2(1 + \phi^2 + s^2)^{\zeta-1} ds \quad (131)$$

rewritten as

$$J_\zeta(x; \phi^2) = x(1 + \phi^2 + x^2)^\zeta - 2\zeta (J_\zeta(x; \phi^2) - (1 + \phi^2)J_{\zeta-1}(x; \phi^2)) \quad (132)$$

Putting together index-shifted Eq. (130) and rearranged Eq. (132) yields the following coupled recurrence relation

$$\begin{cases} I_\zeta(x; \phi^2) &= I_{\zeta-1}(x; \phi^2) + J_{\zeta-1}(x; \phi^2) \\ J_\zeta(x; \phi^2) &= \frac{1}{1+2\zeta} [x(1 + \phi^2 + x^2)^\zeta + 2\zeta(1 + \phi^2)J_{\zeta-1}(x; \phi^2)] \end{cases} \quad (133)$$

The I_ζ of interest (half-integer $\zeta \geq 1$) can then be constructed from the initial evaluations of the integrals (128) and (130). As the lowest for value for n is $n = 0$, the terms starting from $n = -1$ must be evaluated. Since $\zeta = \frac{n}{2} + 1$ the starting contributions are then considered

at $\zeta = 1$ and $\zeta = 0.5$. The $\zeta = 1$ evaluations are simply integrated, whilst the $\zeta = 0.5$ evaluations can be found via symbolic integration or otherwise.

$$\begin{cases} I_1(x; \phi^2) &= x \\ J_1(x; \phi^2) &= \frac{1}{3}x^3 + (1 + \phi^2)x \end{cases} \quad (134)$$

and

$$\begin{cases} I_{0.5}(x; \phi^2) &= \log \left(x + \sqrt{1 + \phi^2 + x^2} \right) - \frac{1}{\phi} \tan^{-1} \left(\frac{\phi x}{1 + \phi^2 + \sqrt{1 + \phi^2 + x^2}} \right) \\ J_{0.5}(x; \phi^2) &= \frac{1}{2} \left(x \sqrt{1 + \phi^2 + x^2} + (1 + \phi^2) \log \left(x + \sqrt{1 + \phi^2 + x^2} \right) \right) \end{cases} \quad (135)$$

Here $\phi = \sqrt{\phi^2}$ (positive square root) without loss of generality.

It is noted that the formulae involving $\tan^{-1}$ have been rearranged using the arctangent addition formula in order to avoid singularities in the argument arising where $\phi = 0$. In the rewritten form, the singularity at $\phi = 0$ is simply handled by taking the limit $\phi \rightarrow 0$

$$I_{0.5}(x; 0) = \log \left(x + \sqrt{1 + x^2} \right) - \frac{x}{1 + \sqrt{1 + x^2}} \quad (136)$$

which follows easily from $\tan^{-1}(x) \sim x$ for $|x| \ll 1$.

It is important to note that while closed-form expressions for I_ζ of interest can now be constructed explicitly, they will generally be cumbersome and involved. A different, in particular a much more efficient approach, is highly recommended in implementations. First, (133) is applied "pointwise" in the following way. For a general polynomial W , and given x, ϕ^2 (e.g. as in (129)), $I_1(x; \phi^2), J_1(x; \phi^2)$ if W has any even degree terms, and $I_{0.5}(x; \phi^2), J_{0.5}(x; \phi^2)$ are evaluated if W has any odd degree terms. Any other required I_ζ (with the maximum ζ dictated by the degree of the polynomial W) are evaluated by repeatedly applying the recurrence relations (133) to those initial values while treating x and ϕ as constant. In this way, the series of values is generated in a sequence of simple and cheap steps only requiring a small number of multiplications. Conversely, explicitly constructing I_ζ starting from (134) and (135), say, would make each subsequent function increasingly involved and result in a much more computationally expensive evaluation (cf [105, 28]).

Second, note that iterating the first equation in (133) for positive integer k yields

$$\begin{aligned} I_k &= I_1 + (J_1 + \dots + J_{k-1}) \\ I_{k+0.5} &= I_{0.5} + (J_{0.5} + \dots + J_{k-0.5}) \end{aligned} \quad (137)$$

The dependence on x, ϕ^2 (treated as constants) is implicit for simplicity of notation. The above allows for sequentially computing a series of values of I_ζ by only iterating the

self-contained recurrence relation for J_ζ in (133). In this way only a single value of J_ζ (the current one) need be stored in memory.

The case $\rho = 0, \lambda \neq 0$ for H_W^3 (Eq. (94)) is relevant if integration is performed numerically (so that $\frac{0}{0}$ does not occur in implementations), but otherwise in the closed-form solution is effectively handled by the limiting case $\phi \rightarrow 0$ in Eq. (136).

The alternate case, however, $\lambda = 0, \rho \neq 0$ does need to be handled carefully. In this case $\gamma_j^{(i)} \equiv 0$ since λ multiplies the first integral in (102) (more generally $\gamma_j \equiv 0$ for $\lambda = 0$). With the same parameterisation and notation, direct substitution of $H_W^{(2)}(0; \lambda)$ (Eq. (96)) into Eq. (106) yields

$$\begin{aligned} \gamma_j^{(i)} &= 0 \\ \nabla \gamma_j^{(i)} &= -\mathbf{n} \mu \sum_{n+2} \frac{a_n}{n+2} [J_{\frac{n}{2}}(t_2; \mu^2 - 1) - J_{\frac{n}{2}}(t_1; \mu^2 - 1)] \end{aligned} \quad (138)$$

In the original definition (130) of J_ζ the second argument was non-negative, which is not guaranteed in the above: $\phi^2 = \mu^2 - 1$ means that the numerator of the integrand I_ζ (Eq. (128)) is not guaranteed to be greater than 0. However, J_ζ defined as the integral in (130) is well-defined for the second argument ≥ -1 , which makes the notation in (138) appropriate. In this case only the self-contained recurrence for J_ζ in (133) (second line) is used, together with $J_{0.5}$ in (135) and the starting value

$$J_0(x; \mu^2 - 1) = x \quad (139)$$

where now the recurrence for even n starts from $\zeta = 0$ (not $\zeta = 1$ as before).

4.5 Implementation in an SPH context

Now having access to closed-form solutions of γ and $\nabla \gamma$, their calculation in an SPH framework is addressed, and algorithms which can be exported to any SPH implementation are given.

By exploiting the assumptions necessary for the kernel function in SPH (finite spherical support and normalisation to unity), we can write the Eqs (69) (66) as

$$\gamma_a = 1 + \sum_j \left(\gamma_j - \text{sgn}(\gamma_j) \frac{A_j}{A_0} \right) \quad (140a)$$

$$\nabla \gamma_a = \sum_j \nabla \gamma_j \quad (140b)$$

where A_0 denotes the area of the unit sphere ($A_0 = 2\pi$ in two dimensions, $A_0 = 4\pi$ in three dimensions), γ_j and $\nabla \gamma_j$ denote contributions from all non-trivial boundary regions

within Ω_a , and A_j is the area of the projection of each non-trivial boundary region onto the unit sphere centered at $\mathbf{r}_a$, such that A_j/A_0 represents the fraction of the total area obscured by these non-trivial boundary regions. The index j runs over all contributing linear (two dimensions) or planar (three dimensions) boundary regions within Ω_a . The sign factor $\text{sgn}(\gamma_j)$ guarantees the correct result in particular in cases when the domain of integration is not star-convex with respect to $\mathbf{r}_a$, and $\text{sgn}(0) := 1$.

In essence, Eq. (140a) breaks down the integral over a sphere (circle in 2D) truncated by intersecting boundaries as follows. Starting from the total spherical integral (1), the relative spherical areas obscured by the truncating boundaries (A_j/A_0) are subtracted to isolate the unaffected volume of the sphere[†] ($1 - \sum_j \frac{A_j}{A_0}$). Finally, the volume integrals up to the truncating boundaries are calculated piecewise and summed ($+\sum_j \gamma_j$). This leaves the total remaining volume up to and around the truncating boundaries. This process is consistent regardless of more complicated intersections which are multi-faceted and not star-convex with respect to the centre of the sphere, as long as the definitions and directions are handled properly ($\text{sgn}\gamma_j$). Furthermore, it allows us to maximise the use of easily calculated analytical geometric expressions, and isolates any tricky integration to the γ_j term.

In the rest of this section, the focus is on efficient evaluation of γ_j , $\nabla\gamma_j$ and A_j in the context of implementation in an SPH code. It is assumed that three-dimensional boundaries consist of convex polygons and can discretised into piecewise-planar elements $\partial\Omega_j$. Since the equations have been presented and spherical boundaries have a straight-forward closed-form solution, the focus is on constructing the geometry of the system appropriately so that the geometric constructs involved and $t_{1,2}$ can be efficiently evaluated to solve Eqs (108) and (129).

4.5.1 Evaluating γ_j , $\nabla\gamma_j$, A_j

Consider a convex planar boundary element $\partial\Omega_j$ in plane P spanned by its corners $\mathbf{x}_{1,\dots,n}$. The ordering of the corners is such that the element is followed anti-clockwise with respect to the unit outward normal of $\partial\Omega_j$, $\mathbf{n}$, which is assumed to be provided. Due to the finite size of the spherical support Ω_a , the general non-trivial intersection $\partial\Omega_j \cap \Omega_a$ will be a planar element with its edge described by a combination of straight lines and circular arcs.

First, λ , the signed perpendicular distance to P , is computed from

$$\lambda = \mathbf{n}^T(\mathbf{x}_1 - \mathbf{r}_a) \quad (141)$$

where $\mathbf{x}_1$ can be replaced by any other corner of the polygon. The element contributes only

[†]The equivalency between relative area and relative volume owes to the volume normalisation to 1

if $(2h)^2 - \lambda^2 > 0$ (else the plane of the boundary element lies outside of the support), which is assumed, and is skipped otherwise. The intersection $\Omega_a \cap P$ is a circle centered at $\mathbf{x}_P$ with radius

$$\rho_{\perp} = \sqrt{(2h)^2 - \lambda^2} \quad (142)$$

Next, consider a single edge l_i of the polygon, spanned by $\mathbf{x}_{i,i+1}$ in this order ($\mathbf{x}_{n+1} = \mathbf{x}_1$). For this edge we compute

$$\hat{\mathbf{e}} = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|} \quad (143)$$

and

$$\mu = \mathbf{n}^T((\mathbf{x}_i - \mathbf{r}_a) \times \hat{\mathbf{e}}) \quad (144)$$

as based on Eq. (103).

The edge has a non-trivial contribution only if $\rho_{\perp}^2 - \mu^2 > 0$ (else P intersects the support but the line containing the edge l_i does not). Also note that $\mathbf{r}_P$ is on the left of l_i in plane P if $\mu \geq 0$ and vice versa. For a contributing edge, the parameterisation is evaluated as

$$\begin{aligned} t_{1,2} &= \hat{\mathbf{e}}^T(\mathbf{x}_{i,i+1} - \mathbf{r}_a) \\ t_{1,2} &\leftarrow \min \left(\sqrt{\rho_{\perp}^2 - \mu^2}, \max \left(-\sqrt{\rho_{\perp}^2 - \mu^2}, t_{1,2} \right) \right) \end{aligned} \quad (145)$$

where the second operation is a truncation which limits the parameterisation to the boundary of the support in the case that the edge extends further out. The edge contributes only if $t_2 > t_1$ (else the edge lies completely outside the support and the truncation sets $t_1 = t_2$) and is skipped otherwise. This parameterisation and truncation combination covers all possible straight line intersections with the spherical support. The contributions to $\gamma_j, \nabla \gamma_j$ from the edge l_i , written $\gamma_j^{(i)}, \nabla \gamma_j^{(i)}$, are evaluated as in (129) with the truncated parameterisation (145) and λ, μ as in (141), (144). This handles the straight line intersections (Sec. 4.2.1). For circular arc intersections (Sec. 4.2.2), it is also necessary to calculate the angle $\psi^{(i)}$ that each segment l_i subtends from the center $\mathbf{r}_P$ in the plane P . $\psi^{(i)}$ is found by defining the edges of the line segment in the appropriately oriented coordinate frame centered at $\mathbf{r}_P$ as (μ, t_1) and (μ, t_2) , and taking their dot product. This is given by the expression

$$\psi^{(i)} = \cos^{-1} \left(\frac{\mu^2 + t_1 t_2}{\sqrt{(\mu^2 + t_1^2)(\mu^2 + t_2^2)}} \right) \quad (146)$$

Only the case $\mu = t_1 t_2 = 0$ is not well defined given the current information. This case will be discussed in more detail at the end of this section.

After all sides $l_{1,\dots,n}$ are considered, the required terms are initialised

$$\begin{aligned}\gamma_j &= \sum_{i^*} \gamma_j^{(i^*)} \\ \nabla \gamma_j &= \sum_{i^*} \nabla \gamma_j^{(i^*)} \\ \psi &= - \sum_{i^*} \text{sgn}(\mu^{(i^*)}) \psi^{(i^*)}\end{aligned}\tag{147}$$

where the index i^* runs only over contributing segments (i.e. those with $\mu^2 + \lambda^2 < R^2$) and $\mu^{(i^*)}$ is the value of μ as defined in (144) along the (contributing) edge l_{i^*} . The sign function in the above is defined at zero as $\text{sgn}(0) = 1$. The above values of $\gamma_j, \nabla \gamma_j$ are not yet exact, since they miss contributions from potential circular arcs along C_j (Sec. 4.2.2).

First, if and only if along all contributing straight edges $\mu \geq 0$ then ψ is updated according to

$$\psi \leftarrow \psi + 2\pi\tag{148}$$

which indicates that $\mathbf{r}_P \in \partial\Omega_j$ (if $\partial\Omega_j$ is a convex polygon). In order to account for circular arcs, the Eqs (108) are used to update

$$\begin{aligned}\gamma_j &\leftarrow \gamma_j + \lambda \psi \rho_\perp^2 H_W^{(3)}(\rho_\perp; \lambda) \\ \nabla \gamma_j &\leftarrow \nabla \gamma_j - \mathbf{n} \psi \rho_\perp^2 H_W^{(2)}(\rho_\perp; \lambda)\end{aligned}\tag{149}$$

This now correctly evaluates the contributions $\gamma_j, \nabla \gamma_j$ to be used in (140b).

The process described above is visualised in Fig. 15, which illustrates how the boundary integrals are appropriately decomposed for a variety of intersections of a triangular $\partial\Omega_j$ and the circle defined by r_P in the plane of intersection P .

The last remaining part of the solution is the evaluation of the projection area A_j of $\partial\Omega_j^{(a)} = \partial\Omega_j \cap \Omega_a$ onto the unit sphere centered at $\mathbf{r}_a$. The projection area $\partial\Omega_j^{(a)}$ is evaluated by splitting the region into a convex polygon and a number of sections of a circle. The polygon is formed from the edges of C_j which pass within the spherical support and the chords on the boundary of the support which connect truncated edges which pass partially outside. The points at the intersections of C_j and $\partial\Omega_a$ where the boundary element edges have been truncated lie on the circle O_P in plane P with radius $\rho_\perp$ centered at $\mathbf{r}_P$, and thus can be connected by arcs of a circle to generate a larger and smaller circular segment. The area of interest A_j is then equal to the area of the projection of the polygon and the sum of the projected areas of the appropriate segments of the circle O_P constructed in this way. A visualisation of this can be seen in Fig. 16.

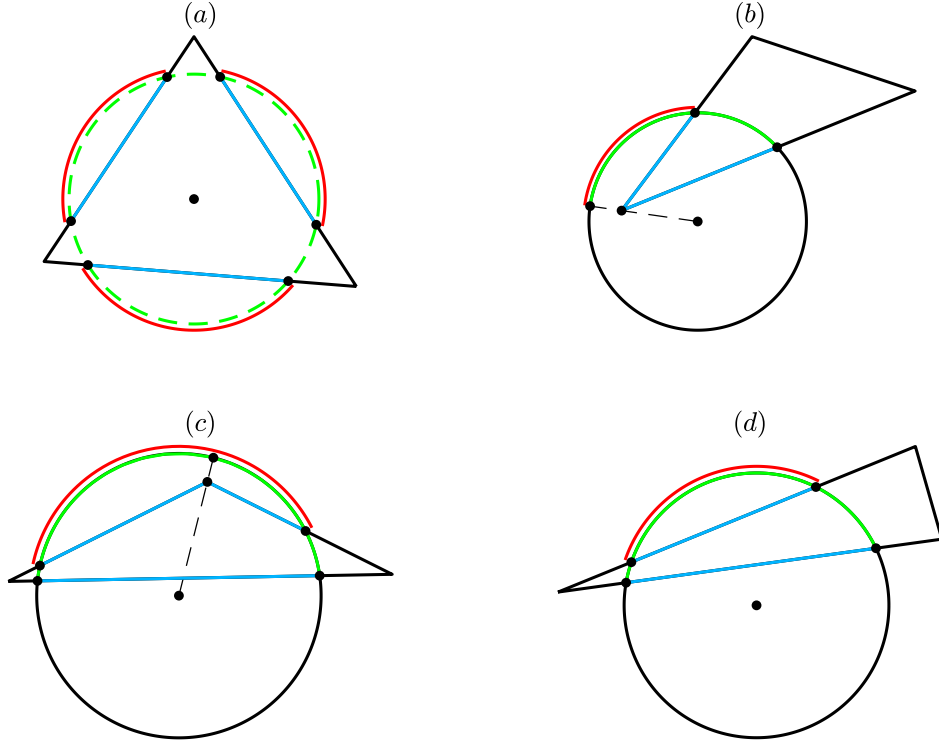


Figure 15: Example intersections of the circle in plane P centered at $\mathbf{r}_P$, and a triangular element $\partial\Omega_j$. These typify example planar intersections that may be encountered when a triangular element intersects a spherical boundary. A full 3D representation of this but with quadrilateral elements can be seen in Fig. 16. The blue straight lines are integrated over, and the relevant end points are constructed based on the parameterisation $t_{1,2}$. The arc length of the projection of each segment is either added to (green) or subtracted from (red) ψ depending on the sign of $\mu^{(i)}$. In (a), the green dashed line indicates that ψ was initialised to 2π since $\mathbf{r}_P \in \partial\Omega_j$ (Eq. (148)). After all line segments are investigated, ψ is the sum of the arc lengths of all the segments to be integrated over.

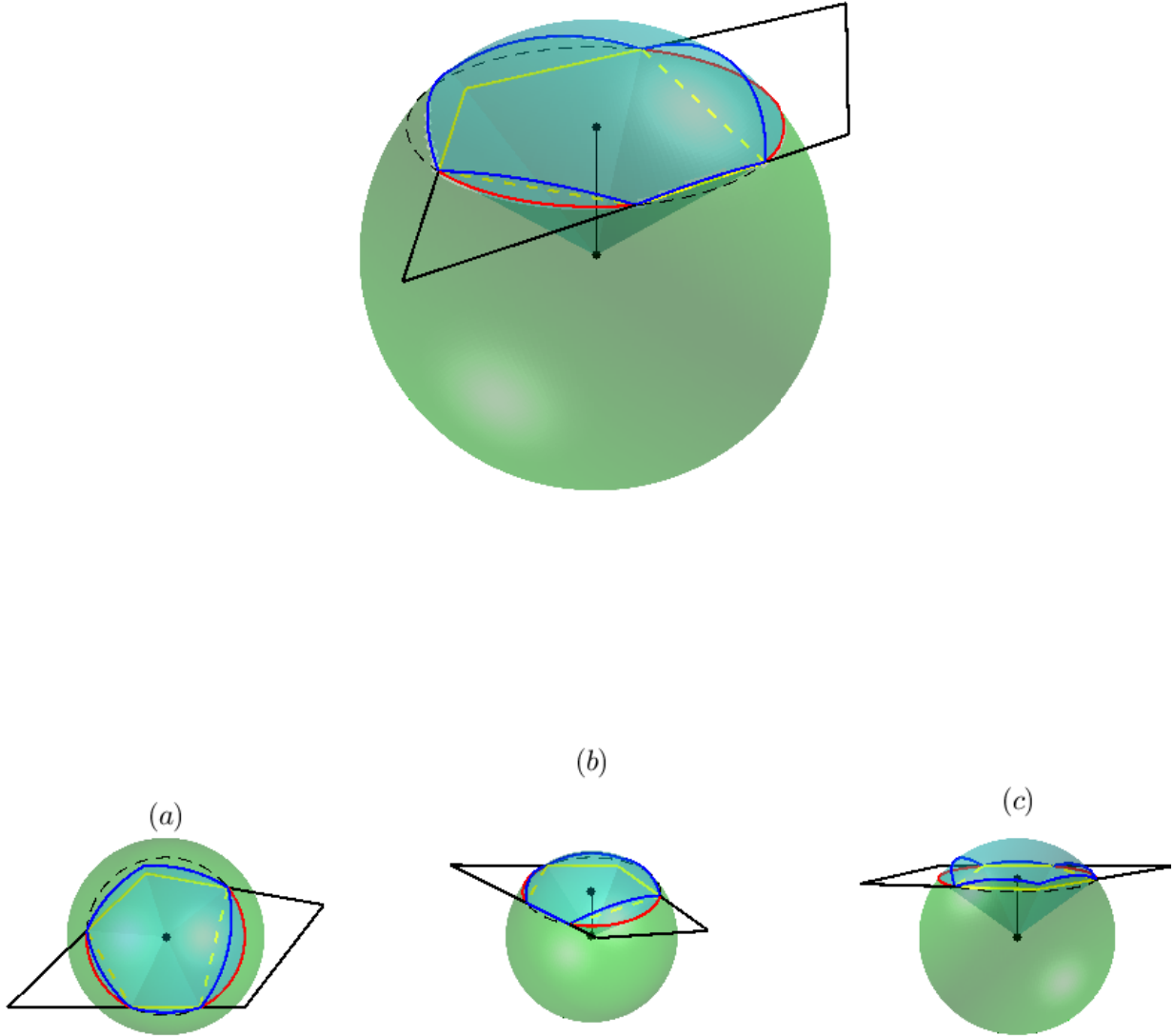


Figure 16: Visualisation of the three-dimensional problem where a particle (green) with radius $R = 1$ (so that $\partial\Omega_a$ is a unit sphere) interacts with a quadrilateral boundary element $\partial\Omega_j$. This is a 3D visualisation of the equivalent diagrams shown in Fig. 15, but with a quadrilateral element instead of the typical triangular element. The contour C_j of $\partial\Omega_j \cap \Omega_a$ is composed of three straight lines (yellow, solid) and two arcs (red) with their end points connected by yellow dashed lines (also see views (a) – (c) for more detail). The projection area A_j (light blue) is the sum of the projection areas of the pentagon bounded by all yellow segments (dark blue contour, cf $A^{[C]}$ in (153)) and two sections of the circle bounded by red arcs and dashed yellow lines (here two spherical regions between red and blue arcs, $A^{(s)}$ in (151)).

The corners of the polygon can be constructed from the parameterisations (145) along the contributing edges, and sections of the circle can be detected and characterised by investigating if (145) was truncated and whether it was truncated from above or below (min/max function).

The projection area of the smaller circular segment of O_P bounded by the chord with length $2d$ is given by

$$A(d) = \begin{cases} \pi, & \lambda = 0, d = \rho_{\perp} \\ \pi - 2 \left(\cos^{-1} \left(\frac{d|\lambda|}{\rho_{\perp} \sqrt{R^2 - d^2}} \right) + \frac{|\lambda|}{R} \sin^{-1} \left(\frac{d}{\rho_{\perp}} \right) \right), & \text{otherwise} \end{cases} \quad (150)$$

This result can be found using spherical trigonometry and realising that the area of interest is the area of the digon bounded by a geodesic (the projection of the chord length $2d$) and a line of constant latitude (the circle O_P of radius $\rho_{\perp}$). Relevant polar triangles can also be examined to find the area of this digon. A brief derivation of this result can be found in online discussion [1].

The result for $\lambda = 0, d = \rho_{\perp}$ is also provided to avoid a computation of $\frac{0}{0}$.

Whether the larger or smaller circular segment need be projected depends on the orientation of the edges and associated intersection boundary. Thus, the above result (150) can be expanded to provide an expression for the projection area of either the larger or smaller segment of O_P appropriately. Given the chord connecting truncated edges is spanned by $\mathbf{y}_{1,2}$ in this order, the expression for the projection area of either the larger or smaller segment can be given by

$$A^{(s)}(\mathbf{y}_1, \mathbf{y}_2) = (1 + \sigma)A(\rho_{\perp}) - \sigma A \left(\frac{\|\mathbf{y}_1 - \mathbf{y}_2\|}{2} \right) \quad (151)$$

$$\sigma = \text{sgn} \left(\mathbf{n}^T ((\mathbf{x}_a - \mathbf{y}_1) \times (\mathbf{y}_2 - \mathbf{y}_1)) \right)$$

Importantly here, σ is positive if the chord goes around $\mathbf{r}_P$ in the anti-clockwise fashion (corresponding to the smaller area) and vice versa. In the above, either $\text{sgn}(0) = 1$ or the typical convention $\text{sgn}(0) = 0$ can be used.

For the projection area of the polygon, let

$$\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_m] \quad (152)$$

be an array of projections of the corners of an arbitrary convex planar polygon onto the unit sphere centered at $\mathbf{r}_a$ ($\|\mathbf{c}_k\| = 1$ for all $k = 1, \dots, m$). The ordering of $\mathbf{c}_k$ should correspond to an arbitrary consecutive ordering of the corners of the original polygon. The area of the spherical polygon spanned by $\mathbf{c}_k$ is given by

$$A^{[\mathbf{C}]} = (2 - m)\pi + \sum_{k=1}^m \cos^{-1} \left(\frac{\mathbf{c}_{k-1}^T \mathbf{c}_{k+1} - p_k p_{k-1}}{\sqrt{(1 - p_k^2)(1 - p_{k-1}^2)}} \right) \quad (153)$$

$$p_k = \mathbf{c}_k^T \mathbf{c}_{k+1}$$

This result can be found by the standard formula for the area of an n -sided polygon on the unit sphere, which is itself just an extension of Girard's theorem, which relates the area

of a triangle on the unit sphere to the sum of its internal angles. In this application, the internal angles, represented by the $\cos^{-1}(\dots)$ term, are found from an extension of the inner product between adjacent vectors connecting vertices of the polygon on the unit sphere. A more complete discussion of this can be found in [83]. Note that in implementations it is preferred to evaluate just $p_{1,\dots,m}$ first and then use them in the above formula in order to minimise the computational expense (the naive implementation would evaluate each p_k twice). Also observe that necessarily $m \leq 2n$ where n is the number of sides/corners of the boundary element $\partial\Omega_j$; the latter bound may be important in implementations if the amount of memory allocated to $\mathbf{C}$ has to be specified *a priori*.

Many of the terms involved in the calculations presented, particularly (153), produce singularities for certain inputs. Whilst these inputs should in practice be exceedingly rare, experience with this implementation shows that finite machine precision can often lead to situations where the function evaluations do become singular. This is because the formulae presented involve repeated floating point operations which can produce results where the values are indistinguishable to machine precision. This is often the case when an element corner is relatively close (within $\sim 10^{-10}$) to $\partial\Omega_a$. In order to avoid these floating point round-off errors, limits are applied to certain function evaluations where the argument necessarily falls in a given defined domain for the current problem. Here these are $\cos^{-1} : [-1, 1] \rightarrow \mathbb{R}$, $\sqrt{x} : x \geq 0 \rightarrow \mathbb{R}$, $1/x : x \neq 0 \rightarrow \mathbb{R}$, affecting Eqs (150) and (153).

First, in Eq. (153) a limit is applied

$$p_k \leftarrow \max(-1, \min(1, p_k)) \quad (154)$$

which ensures that the argument of $\sqrt{\dots}$ in the denominator of each term in (153) is non-negative.

Second, each term in the sum is modified according to

$$\cos^{-1}\left(\frac{\dots}{\sqrt{\dots}}\right) \rightarrow \cos^{-1}\left(\max\left[-1, \min\left[1, \frac{\dots}{\max(\epsilon, \sqrt{\dots})}\right]\right]\right) \quad (155)$$

where ϵ is a small positive user-specified tolerance on the order of machine precision; in the current implementation $\epsilon = 10^{-14}$ is used. The above modifications directly prohibit singularities from occurring and guarantee that the argument of $\cos^{-1}$ is in the appropriate range $[-1, 1]$. Similar but simpler safeties are used in (150) for the arguments of $\sin^{-1}$ and $\cos^{-1}$. Since both of them are positive and $\rho_{\perp}, R^2 - d^2 > 0$ are already guaranteed when $A(d)$ is evaluated, each argument is only limited from the top via $\min(1, \cdot)$. An explicit mechanism for avoiding singularities in (150) from division by zero can also be used (cf $\max(\epsilon, \cdot)$ in (155)).

An efficient and fully general implementation of this approach for polynomial W is summarised in Algorithm 1 below. The method handles all cases, including those where the particle lies on a boundary face, with the only exceptions when the particle lies directly on or above (perpendicularly) a boundary corner. These cases are predicated on the previously mentioned case $\mu = t_1 t_2 = 0$, which is a necessary condition for the problematic cases. In one case, this corresponds to the fluid particle being precisely at a corner of $\partial\Omega$. When the particle lies directly on the boundary corner, $\lambda = 0$ as well, and the projection area of the element A_j onto the unit sphere is not well defined. Evaluating the appropriate contribution requires information on neighbouring elements. A simple safety mechanism is to move the particle away from the corner by a tiny amount in the negative mean direction of the connected boundary normals for the purpose of the calculation, such that there is no singularity but that the calculation result remains accurate to the configuration. This can be implemented as

$$\tilde{\mathbf{r}}_a = \begin{cases} \mathbf{r}_a - \epsilon \hat{\mathbf{n}}, & \mu = \lambda = t_1 t_2 = 0 \\ \mathbf{r}_a, & \text{otherwise} \end{cases} \quad (156)$$

where $\tilde{\mathbf{r}}_a$ is the temporary shifted particle position, ϵ is a arbitrary constant on the order of the finite machine precision, and $\hat{\mathbf{n}}_i$ is the mean unit normal of the boundary elements connected to the vertex in consideration. In practice, it is very unlikely that $\mathbf{r}_a \in \partial\Omega$ as long as repulsion and leaking mechanisms are appropriately implemented.

The second case however, where $\mu = t_1 t_2 = 0$ but the particle is away from the walls (and hence $\lambda \neq 0$), may happen more frequently and cannot be guaranteed to avoided by repulsion/leak handling. This case occurs when $\mathbf{r}_a$ lies directly above the corner of an element in a direction perpendicular to one of the connected boundary elements. In this instance, the particle position can again be shifted by a very small amount in the direction away from the corner as in Eq. (156), or more simply, without knowledge of connected segments, can be shifted by a very small amount in the direction from the vertex to the center of the problematic boundary element - $\mathbf{x}_{ic} = \left(\frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3}{3} - \mathbf{x}_i\right)$. This can be implemented as

$$\tilde{\mathbf{r}}_a = \begin{cases} \mathbf{r}_a - \epsilon \hat{\mathbf{x}}_{ic}, & \mu = t_1 t_2 = 0, \lambda \neq 0 \\ \mathbf{r}_a, & \text{otherwise} \end{cases} \quad (157)$$

After these appropriate shifts to $\mathbf{r}_a$ are applied if necessary, μ , λ , and t_1 and t_2 should all be recalculated. Calculations will then be non-singular and the algorithm can be continued as normal.

The correct handling of $\mu, \lambda = 0$ in implementations is crucial, even if the particle position

$\mathbf{r}_a$ is always assumed to be away from walls. Indeed, for $\mathbf{r}_a$ along the line containing a straight edge of an element within the particle support (but away from the edge itself and $\partial\Omega$ generally), one has $\mu = \lambda = 0$ but still lies within the support and requires calculation of the appropriate terms.

Algorithm 1 Pseudocode for the three-dimensional algorithm for polynomial W with the support radius R . All of γ_j, A_j and $\nabla\gamma_j$ are computed in the same loop. An additional switch can be added after line 15, so that $t_1 t_2 \leq 0 = \mu$ raises an error.

Require: Appropriately oriented corners of a convex polygon $\partial\Omega_j, \mathbf{x}_1, \dots, \mathbf{x}_n$ and the outward unit normal $\mathbf{n}$

Ensure: Element contributions to $\gamma/\nabla\gamma$: $\gamma_j/\nabla\gamma_j$, and the area A_j to be used in (140a)/(140b)

```

1:  $\gamma_j \leftarrow 0$ 
2:  $\nabla\gamma_j \leftarrow \mathbf{0}$ 
3:  $A_j \leftarrow 0$ 
4: Evaluate  $\lambda = \mathbf{n}^T(\mathbf{x}_1 - \mathbf{x}_a)$ 
5: if  $(2h)^2 > \lambda^2$  then
6:    $\mathbf{y}_1 \leftarrow []$ ,  $\mathbf{y}_2 \leftarrow []$  //Vertices of truncated intersecting edges
7:    $\mathbf{C} \leftarrow []$  //Projection of intersecting vertices onto unit circle
8:    $\psi \leftarrow 0$ ,  $L_0 \leftarrow \text{true}$ 
9:   Evaluate  $\rho_\perp = \sqrt{R^2 - \lambda^2}$ 
10:  for  $i = 1, \dots, n$  do
11:    Evaluate  $\hat{\mathbf{e}}$  and  $\mu$  as in (143) and (144)
12:    if  $\mu < 0$  then  $L_0 \leftarrow \text{false}$ 
13:    if  $\rho_\perp^2 > \mu^2$  then
14:      Evaluate and truncate  $t_{1,2}$  as in (145). During the process evaluate:
15:       $L_1 \leftarrow "t_1 \text{ was truncated from the bottom in (145)"}
16:       $L_2 \leftarrow "t_2 \text{ was truncated from the top in (145)"}
17:      if  $t_2 > t_1$  then
18:        Evaluate  $\gamma_j^{(i)}/\nabla\gamma_j^{(i)}$  as in (129) (or (138) if  $\lambda = 0$ ) and  $\psi^{(i)}$  according to (146)
19:        if  $\mu \geq 0$  then  $\psi^{(i)} \leftarrow -\psi^{(i)}$ 
20:         $\gamma_j \leftarrow \gamma_j + \gamma_j^{(i)}$ 
21:         $\nabla\gamma_j \leftarrow \nabla\gamma_j + \nabla\gamma_j^{(i)}$ 
22:         $\psi \leftarrow \psi + \psi^{(i)}$ 
23:        if  $L_1$  then
24:           $\mathbf{z} = \mathbf{x}_i + [t_1 - \hat{\mathbf{e}}^T(\mathbf{x}_i - \mathbf{x}_a)]\hat{\mathbf{e}}$  //Create appropriate chord vertex based on truncation
25:          if  $\mathbf{y}_1 \neq []$  then
26:             $A_j \leftarrow A_j + A^{(s)}(\mathbf{y}_1, \mathbf{z})$  with  $A^{(s)}(\cdot, \cdot)$  as in (151)
27:          else
28:             $\mathbf{y}_2 \leftarrow \mathbf{z}$ 
29:          end if
30:          Append  $\mathbf{C} \leftarrow [\mathbf{C}, \frac{\mathbf{z} - \mathbf{x}_a}{\|\mathbf{z} - \mathbf{x}_a\|}]$ 
31:        end if
32:         $\mathbf{z} = \mathbf{x}_i + [t_2 - \hat{\mathbf{e}}^T(\mathbf{x}_i - \mathbf{r}_a)]\hat{\mathbf{e}}$  //Create appropriate chord vertex based on truncation
33:        Append  $\mathbf{C} \leftarrow [\mathbf{C}, \frac{\mathbf{z} - \mathbf{x}_a}{\|\mathbf{z} - \mathbf{x}_a\|}]$ 
34:        if  $L_2$  then  $\mathbf{y}_1 \leftarrow \mathbf{z}$ 
35:      end if
36:    end if
37:  end for
38:  if  $L_0$  then
39:    if  $\psi = 0$  then  $A_j \leftarrow 2A(\rho_\perp)$  with  $A(d)$  as in (150)
40:     $\psi \leftarrow \psi + 2\pi$ 
41:  end if
42:  Update  $\gamma_j$  and  $\nabla\gamma_j$  as in (149)
43:  if  $\mathbf{y}_2 \neq []$  then  $A_j \leftarrow A_j + A^{(s)}(\mathbf{y}_1, \mathbf{y}_2)$  with  $A^{(s)}(\cdot, \cdot)$  as in (151) //Projection area of segment
44:  if  $\text{size}(\mathbf{C}) \geq 3$  then  $A_j \leftarrow A_j + A^{[\mathbf{C}]}$  with  $A^{[\mathbf{C}]}$  as in (153) //Projection area of polygon
45: end if$$ 
```

The calculations and algorithm presented here are mathematically closed-form, and as such are inherently completely accurate outside of machine-precision rounding errors. For validation of the implementations, simple test cases can be used to confirm the accurate calculation of γ from algorithm 1 for known results.

The case of a single particle in the vicinity of a standard corner boundary was modelled in both 2D [†] and 3D, shown in Fig. 17.

As a single particle orthogonally approaches a regularly cornered boundary, γ_a will decrease as the area/volume is obstructed by said corner. At the point where the corner of the boundary meets the center of the particle support, the expected result for γ_a can be geometrically understood as just the percentage of the particle that is unobstructed by the support (due to the symmetry of the situation). In this case, one quadrant of the circle/sphere. As such, in 2D, $\gamma_a = 0.75$ as the corner obstructs 1/4 of the particle support, whilst in 3D, $\gamma_a = 0.875$ as the corner obstructs 1/8 of the particle support. This result is reproduced exactly by the proposed algorithm 1.

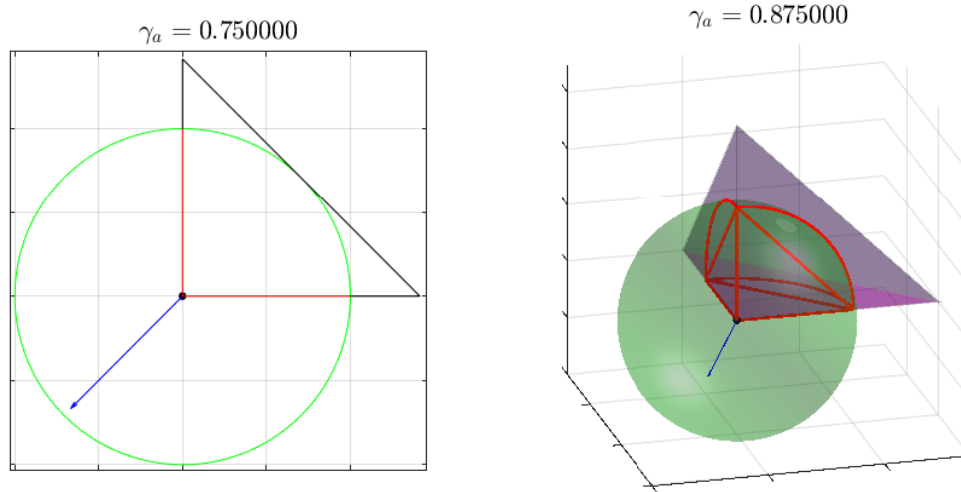


Figure 17: γ_a for the problem of a particle approaching a corner. The known analytical result is reproduced exactly, as expected. The blue vector line represents $\nabla\gamma_a$.

The algorithm and calculations can also be tested for their robustness in particularly complicated boundary intersections. These results do not serve as an accuracy validation since they are not known analytical solutions, but do confirm the algorithm's ability to handle all types of geometries. Example test cases are shown in Fig. 18. Note that due to the asymmetry and complexity of the problem, γ_a cannot be equated to the volume obscured

[†]The 2D algorithm, though not presented here, is functionally equivalent and full details can be found in [84] as mentioned.

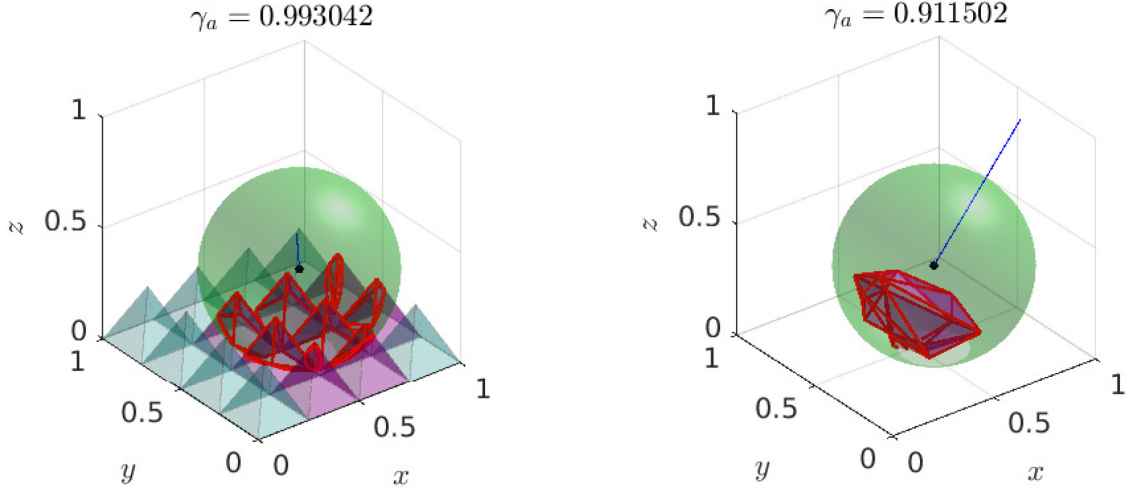


Figure 18: Test problems of a particle support intersecting multi-faceted complex boundaries. A particle with a quintic Wendland kernel and radius $2h = 0.4$ is centered at position $(0.55, 0.45, 0.3)$ and intersects a complex boundary non-symmetrically. The red highlighted lines of the boundary identify where line integrals are calculated. The second example is able to evaluate γ_a despite the lack of star-convexity of the boundary with respect to the particle position. The blue lines represent $\nabla \gamma_a$.

by the intersection as in the previous example, and instead must be fully calculated using the kernel integration.

5 An improved semi-analytical boundary method (2) - Further considerations and preliminary results

5.1 Governing equations

The semi-analytical framework is defined by the regular SPH implementation presented in Chapter 2, however with the operators for values and derivatives defined as in Sec. 3.3. The standard semi-analytical governing equations involving an appropriately adjusted delta-SPH method from Eq. (36) can then be written as

$$\begin{aligned} \frac{d\rho_i}{dt} = & \frac{1}{\gamma_i} \sum_j V_j \left(\rho_j \mathbf{u}_{ij} \cdot \mathbf{e}_{ij} + 2\delta h c_0 \frac{\rho_{ij}^D}{|\mathbf{r}_{ij}|} \nabla W_{ij} \right) \\ & - \frac{1}{\gamma_i} \sum_w W_{iw} dS \left(\rho_w \mathbf{u}_{iw} \cdot \mathbf{n}_w + 2\delta h c_0 \frac{\rho_{iw}^D}{|\mathbf{r}_{iw} \cdot \mathbf{n}_w|} \right) \end{aligned} \quad (158)$$

$$\begin{aligned} \frac{d\mathbf{u}_i}{dt} = & - \frac{1}{\gamma_i} \left(\sum_j \rho_j V_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \mathbf{e}_{ij} - \sum_w \rho_w \left(\frac{P_i}{\rho_i^2} + \frac{P_w}{\rho_w^2} \right) W_{iw} \mathbf{n}_w dS \right) \\ & + 2\nu \frac{1}{\gamma_i} \left(\sum_j 2 \frac{\mathbf{u}_{ij}}{|\mathbf{r}_{ij}|} V_j \nabla_i W_{ij} - \sum_w 2 \frac{\mathbf{u}_{iw}}{|\mathbf{r}_{iw} \cdot \mathbf{n}|} W_{iw} dS \right) \\ & + \mathbf{g} + \frac{\mathbf{f}_i}{m_i} \end{aligned} \quad (159)$$

These equations are used to update the system variables following a standard time-integration scheme as presented in Sec. 2.7.1. Pressures at the wall particles P_w are assigned based on a simple fluid average as in Eq. (53). This approach of setting the wall particle pressures via a local average of the fluid particles is often used as the most basic approach for ghost particle boundary methods, but it has also become standardised for the semi-analytical boundary method following the work by Ferrand et al. [53].

Here, the above continuity equation (158) includes the delta-SPH density smoothing method as standard. This was done due to its ubiquitous effectiveness within multiple different forms of SPH. To generate the appropriate semi-analytical version of the method, the additional density smoothing term was written as a Laplacian of the density in the continuity equation, generating an additional term $\delta h c_0 \nabla^2 \rho^D$. The semi-analytical formulation of the Laplacian in Eq. (65) could then be applied.

It is noted that most implementations in the literature do not use delta-SPH within the semi-analytical framework. Instead they use the time-independent approach for calculat-

ing density as presented by Ferrand et al. [53]. The time-independent approach calculates density based on particle positions and is thus exact to the system configuration. No integration of the continuity equation is required and the need for density smoothing techniques is avoided due to the improved calculation accuracy. This is analogous to the discussion on density in Sec. 2.6.1, and will be discussed in more detail in the following section 5.2.

With the model defined by Eqs (158) and (159), initial simulations were run to assess its viability. Results of the Kleefsman 3D dam break [81] are presented in the below Fig. 19

The results in Fig. 19 show that while this basic semi-analytical boundary implementation is able to work reasonably well on first glance, it is not without multiple issues. The pressure field is very noisy, especially near corners of the domain. Pressure waves within the flow can also be observed. The free surface exhibits some disturbance and noise, with particles becoming detached from the flow. Most importantly, though not visible in these snapshots, is that the method displays extremely poor containment. Almost all the particles were not contained at the point of fluid impact with the fixed obstacle, and had to be reflected back into the fluid domain as described in Sec. 3.1. By 1.0s of simulation time, around 100 fluid particles were passing outside the fluid domain and having to be reflected back per time step, accounting for roughly 0.25% of the fluid particles being lost per time step. This is unfeasibly large, and if the fluid particles were not reflected back, then around 10% of the fluid particles would have been lost after just the first impact. These observed behaviours motivated improvements to the method.

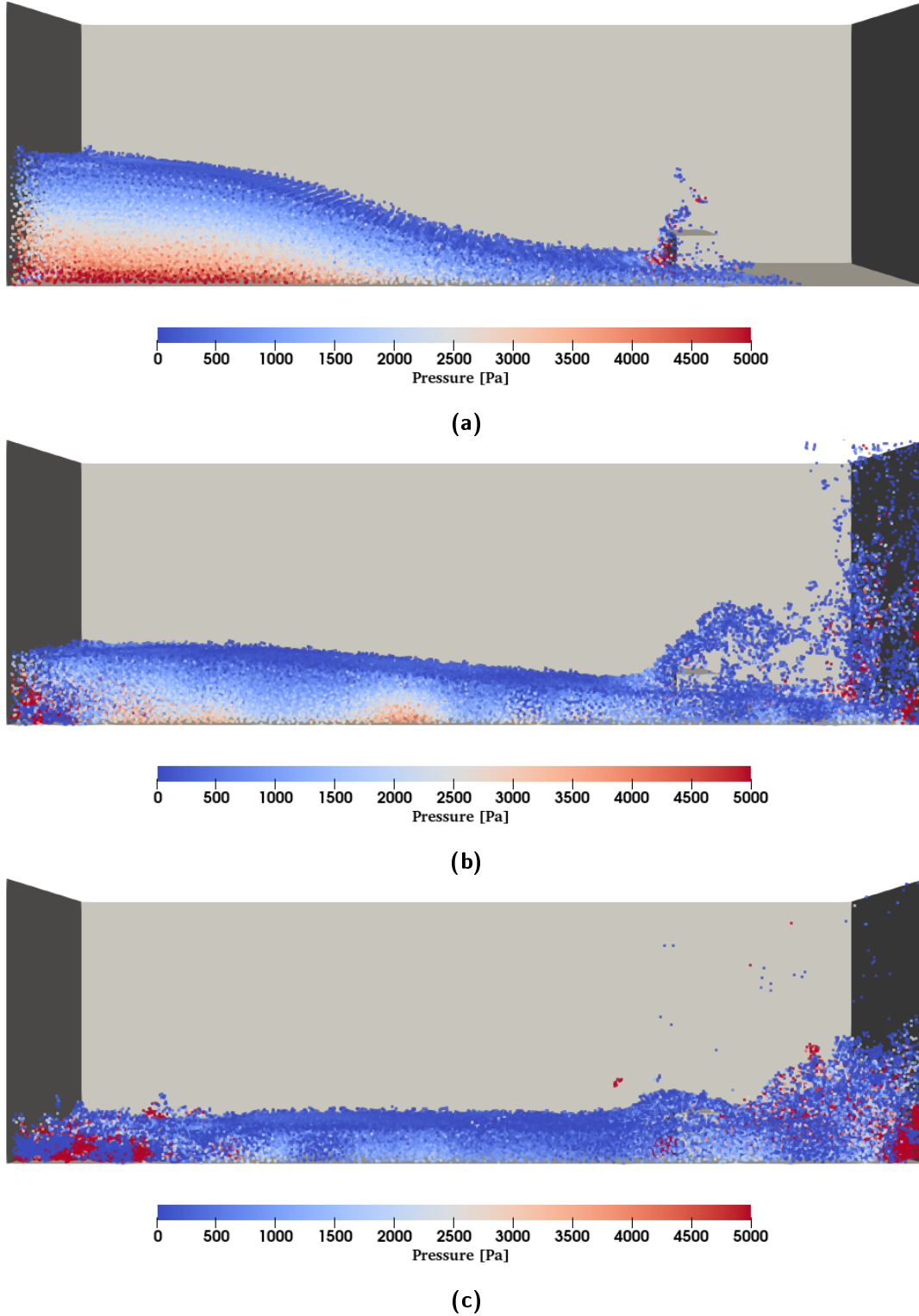


Figure 19: Simulation of the 3D dam break test case as described by Kleefsman [81]. The x domain displayed is from 0 m to 3.22 m, as specified in the original test case. The semi-analytical boundary method following Eqs (158) (159) with no additional adjustments is used. Fluid particle resolution is $\Delta x = H/22$, which is notably less than the standard for the Kleefsman dam break of $\Delta x = H/33$. Snapshots of the pressure field are presented at (a) $t = 0.5$ s, (b) $t = 1.0$ s, and (c) $t = 1.5$ s.

5.2 Further considerations

Firstly, it was noted that the semi-analytical boundary method presented does not hold fluid particles at a distance of Δx from the wall as might be assumed. As discussed with the ghost particle methods in Sec. 3.2, boundary particle positioning is an important aspect to consider in order to produce the presentation of accurate fluid flow near boundaries. It is often useful to run test simulations to identify the spacing generated between wall particles and fluid particles. Initial simulations showed that the semi-analytical boundary method produces a natural repulsion distance of $\Delta x/2$. This appeared consistent over multiple simulation regimes and boundary shapes. For this reason, fluid particles should not be initialised at a distance Δx from the walls as is standard, but rather at a distance of $\Delta x/2$. In this way, fluid particles will start the simulation in a more contained and stable position and avoid generating large oscillations and pressure waves at the start of a simulation which can build noise throughout. This initial particle positioning is equivalent to the discussion in Sec. 2.6.1 on pressure initialisation.

Next, an alternative method of pressure assignment at the wall particles was attempted. The containment issues suggested that the pressures being generated at wall particles may not be enough, especially in such high velocity impact flows as the 3D dam break. To this end, the pressure at the wall was instead assigned following the more complete, normal fluid flux method, as in Eq. 55.

Finally, the particle shifting method presented in Sec. 2.6.3 was adapted for application to the semi-analytical framework and implemented. The current semi-analytical boundary results (Fig. 19) displayed some irregularity in the particle distribution and clustered or sparse regions near wall impact points with large spikes in pressure. Particle shifting is well known to be effective in improving the modelling of high velocity flows involving impacts such as dam breaks [110]. The extension to the semi-analytical method is therefore natural. Two key terms in the particle shifting technique must be adapted:

$$\nabla C_i = \sum_j V_j \nabla W_{ij} \mathbf{e}_{ij} \quad (160)$$

$$\nabla \cdot \mathbf{r} = -\frac{1}{\gamma_i} \sum_j V_j \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \mathbf{e}_{ij} + \frac{1}{\gamma_i} \sum_w \mathbf{r}_{iw} \cdot W_{iw} \mathbf{n}_w dS \quad (161)$$

After these operators are redefined, it is then possible to implement the particle shifting technique as described in Sec. 2.6.3.

These three adjustments of initial fluid particle positioning, normal flux wall pressure assignment, and semi-analytical particle shifting proved to be very effective. These adjust-

ments, in combination with the closed-form implementation of the method presented in Chapter 4 are what refers to the “improved” semi-analytical boundary method described throughout.

Again, initial results of the 3D dam break based now on the improved semi-analytical boundary method are presented in the below Fig. 20. Comparisons can be drawn with the initial implementation in Fig. 19.

The benefits of the improvements to the semi-analytical boundary method are evident when comparing Figs 20 and 19. The pressure field is less noisy throughout, and the effects of extreme spikes in pressure at impact points are limited. No pressure waves are evident. The article distribution remains more regular throughout, even near the corners of walls and impact points, and the free surface remains smooth and consistent with no particles becoming detached. Most importantly perhaps, very few particles are leaked from the domain - roughly 1% of the total particles over the course of the entire simulation - suggesting the containment offered by the normal flux pressure assignment is much improved. Preliminary results of the method tracking the pressure and free surface height are presented in the following section 5.3.

A separate issue with this implementation of the semi-analytical boundary method is mentioned by Ferrand et al. in their seminal work on the semi-analytical boundary method [53]. They found that the time discretised continuity equation in a semi-analytical framework, as in Eq. 158, produces a systematic error which tends to underestimate the density near a boundary. This reduces the containment effect and leads to fluid particles leaking from the domain. This same behaviour was observed within this work. In Fig. 21 a stationary hydrostatic box of fluid with height $H = 0.35$ m was simulated, and left to settle to its steady-state solution. Of course, this should ideally be a $\mathbf{0}$ velocity field, ignoring small fluctuations.

Fig. 21 shows the velocity magnitude field of a hydrostatic box of fluid after it has settled. It can be clearly observed that the highlighted particles with velocity magnitudes greater than the 90th percentile are concentrated at the edges of the box, where the wall boundaries meet. These spurious velocities tend to generate unphysical secondary currents within the hydrostatic fluid. Fluid particles near the boundary edges and corners are circulated upwards and towards the center of the fluid, and central fluid particles are drawn into and up at the boundary edges, creating a net circulating motion at the boundary edges.

The leaking behaviour described by Ferrand et al. where fluid particles are gradually leaked from the simulation at low speeds was also observed. The improvements to the method presented in this section are focused on high velocity, transient flows, and as such do not offer much improvements in the hydrostatic box simulation. The new pressure assignment

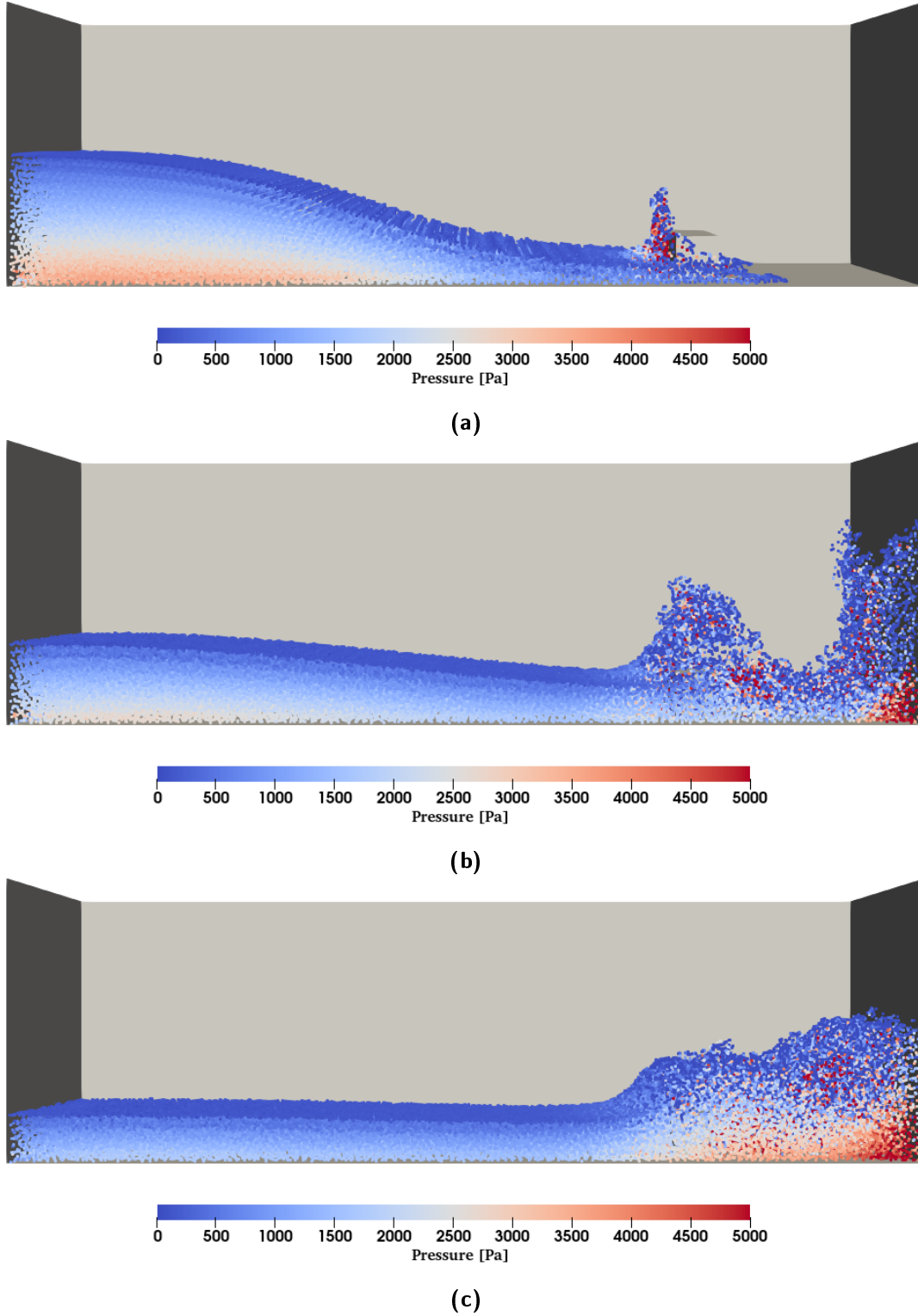


Figure 20: Simulation of the 3D dam break test case as described by Kleefsman [81] using the improved semi-analytical boundary framework outlined in the current section. $\Delta x = H/22$ as in 19. Improvements include a more accurate initial fluid particle positioning, a wall pressure assignment involving normal fluxes, and a particle shifting technique. All simulation parameters and snapshots are the same as in Fig. 19.

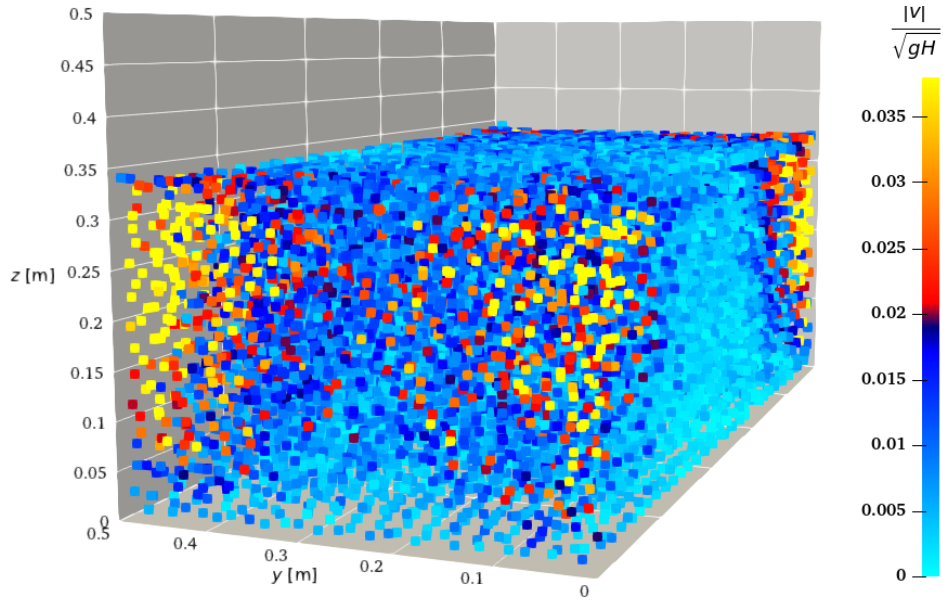


Figure 21: Non-dimensional velocity magnitude of fluid particles settled into a hydrostatic box using the improved semi-analytical boundary method. Dimensions of the box are (1.0, 0.5, 0.5) metres with a fluid height of 0.35 m and a fluid resolution $\Delta x = 0.02$ m. The colour grid is chosen to highlight velocities larger than the 90th percentile value - $|v|/\sqrt{gH} = 0.019$. Bright spots of peaked velocities are found near the boundary edges where wall boundaries meet. Outside of these areas the velocities remain close to 0 as expected.

at the wall utilising the normal fluid flux is likely not enough to limit the leaking since it is a low speed long time period effect where fluxes remain low. Ferrand et al. avoided this by generating a time-independent formulation of the density in the semi-analytical method. The time-independent formulation of density removes all potential integration errors in Eq. (158) and is known to be much more accurate far from free surfaces. Nonetheless, the time-independent formulation of density was not considered within the scope of this work. Implementing a time-independent density formulation can be cumbersome, as it involves storing γ and the kernel function W at multiple time steps - something not commonly done in a typical SPH implementation. Further, this issue appears only to be a problem when particles remain in contact with a boundary for a long time and there is no other dominant motion in the system. This is because the same issue is not observed when simulating a transient problem such as a dam break, as was observed in Fig. 20. Since these transient problems were the original focus of this work, it was not deemed necessary to implement a time-independent continuity equation.

A further issue which may be in part a cause of this density and spurious velocity issue near boundary edges, is the approximation of $\nabla\gamma_{iw} \approx W_{iw}\mathbf{n}_w dS$ utilised in the semi-analytical equations presented in Sec. 3.3. This approximation arose from the need to discretise the surface integral over the boundary. The term it is approximating represents the integral $\int_{\partial\Omega_w} W\mathbf{n}dS$ of the kernel function over a small discretised element of the wall boundary prescribed to a wall particle. This can be equated to $\nabla\gamma_i$ calculated over a small discretised element of the wall boundary. For a given near-wall fluid particle, summing these contributions over all wall particles thus can be equated to $\nabla\gamma_i$ for the entire boundary interaction. From the work in Chapter 4 it is possible to calculate $\nabla\gamma_i$ exactly and as such quantify the quality of the approximation $\nabla\gamma_i \approx \nabla\gamma_e = \sum_w W_{iw}\mathbf{n}_w dS$. The relative error in the magnitude of the approximation $\frac{|\nabla\gamma_e| - |\nabla\gamma_i|}{|\nabla\gamma_i|}$ was calculated for a hydrostatic box of fluid and presented in Fig. 22.

It can be seen in Fig. 22 that errors in the $\nabla\gamma$ approximation are intensely focused around the corners and edges of the solid boundaries. whilst remaining accurate along flat walls. For fluid particles far from the wall boundaries there is no approximation to be made since $\nabla\gamma_i = \mathbf{0}$ identically, and so there is no error. Here $\frac{|\nabla\gamma_e| - |\nabla\gamma_i|}{|\nabla\gamma_i|}$ is set to 0 to avoid division by 0. The errors are particularly large in the boundary corners, where the relative error can reach over 100%, whilst along boundary edges the relative error remains around 30% – 50%. This suggests, perhaps logically, that the further the wall boundary deviates from a flat surface the greater the error in the approximation. Indeed, the approximation is expected to lose accuracy when W_{iw} is not approximately constant over the boundary element. For flat walls the distance to the fluid particle is roughly constant and so W_{iw} , which relies only on

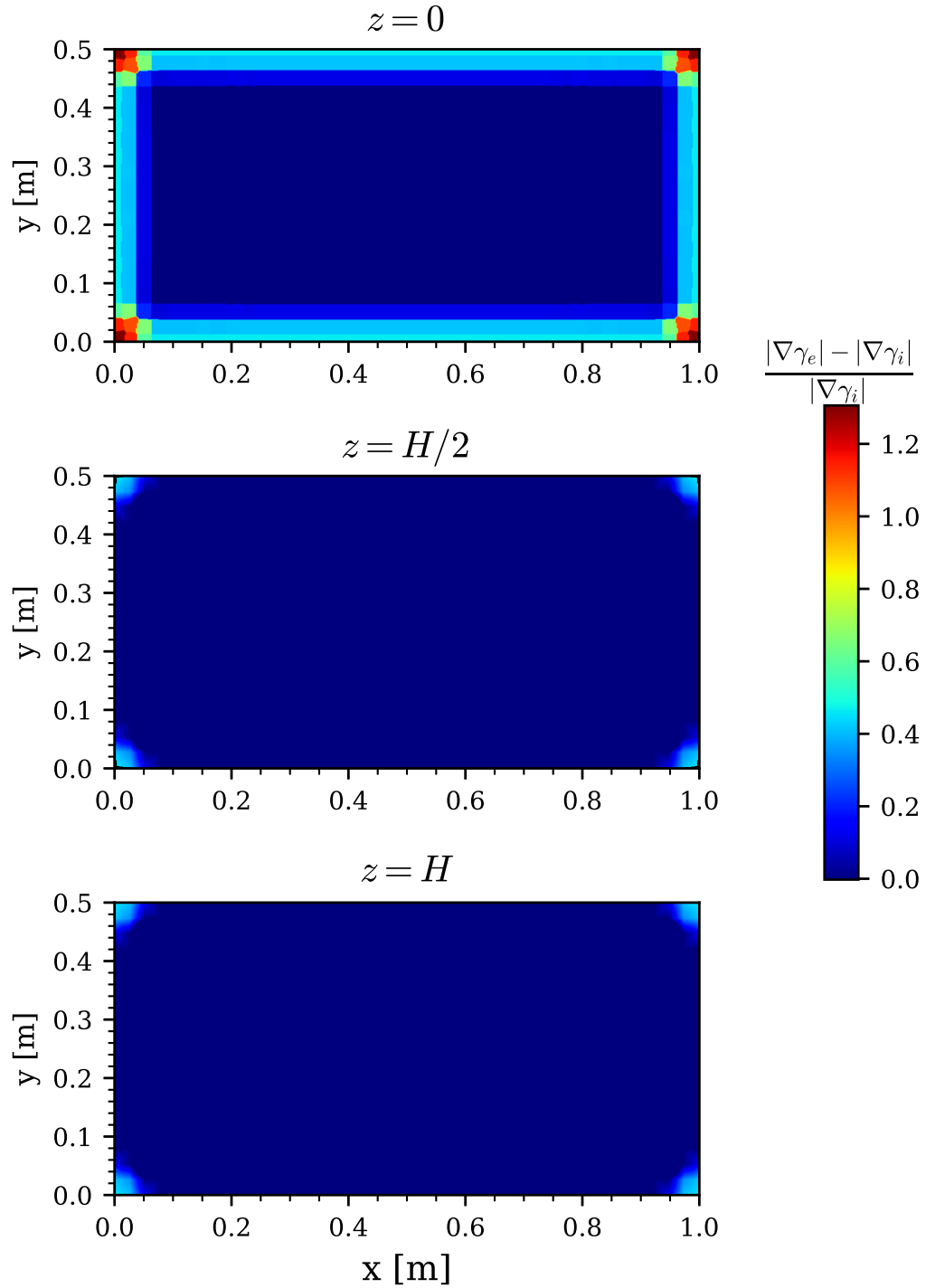


Figure 22: Relative error in the approximation $\nabla\gamma_i \approx \sum_w W_{iw} \mathbf{n}_w dS$ for a hydrostatic box of fluid. Simulation parameters are the same as in Fig. 21. 2D snapshots of the error field in x, y are taken at fixed values of z - at the bottom, middle, and top of the tank. This demonstrates how the relative error field varies both in x, y and z . The error at fluid particles that do not interact with wall boundaries and consequently have $\gamma_i = \mathbf{0}$ are identically set to 0.

distance to the fluid particle, can be approximated as constant. For more complex boundary elements which have greater variations in their distance from the fluid particle (i.e boundary elements at non-normal orientations to the fluid), W_{iw} can no longer be reasonably assumed to be constant. These errors are large enough to merit consideration of the effects on the simulation results, such as the observed spurious velocities near boundary edges and corners in Fig. 21. Large overestimations in the term $W_{iw}\mathbf{n}_w dS$ will lead to overestimations of the acceleration in the equation (159) and may be the cause of the observed spurious velocities and resulting unphysical currents. Ideally this approximation could be avoided and the fully exact solution presented in Chapter 4 could be utilised. More comments will be made on this in the final chapter on future work.

5.3 Preliminary results

In order to assess the accuracy of the improved semi-analytical boundary method, the full time evolutions of the pressure and free surface height at different nodal locations in the 3D dam break were tracked and compared to experimental results. Full details of the simulation set-up including exact locations of pressure nodes and free surface measuring locations can be found in the original work by Kleefsman [81] and online via the validation test case listing by SPHERIC [149]. Simulations presented in this section were run on Imperial College Research Computing Service's high performance computing cluster [140]. All the simulations were performed on AMD 7742 64-Core (128 thread) processors. Compute nodes used ranged from 1 to 3, with the number of MPI processes used ranging from just 4 for the lowest resolutions, up to 356 for the highest resolutions. General benchmarking of computational speed and performance was not carried out.

First, results for pressure and free surface height were compared against the experimental results in Figs 23 and 24. This was done using a standard resolution of $\Delta x = H/22 = 0.025$ m corresponding to roughly 50,000 fluid particles. This is lower than the resolution listed in the validation test case outline of $\Delta x = H/22$, but was standard for many of the simulations completed during this work. Other simulation parameters followed the standard definitions in the SPHERIC test case [149].

It is observed that generally the results for both pressure and free surface height in Figs 23 and 24 are satisfactory in comparison to the experimental data. The overall evolution as well as the timing and occurrence of waves within the system are well captured. One issue does appear to be consistent overestimation of the pressure throughout the simulation, but particularly so during wave impacts against the obstacle. Also, the downstream pressure nodes $P5$ and $P7$ produce very noisy pressure fields. This is likely because these node

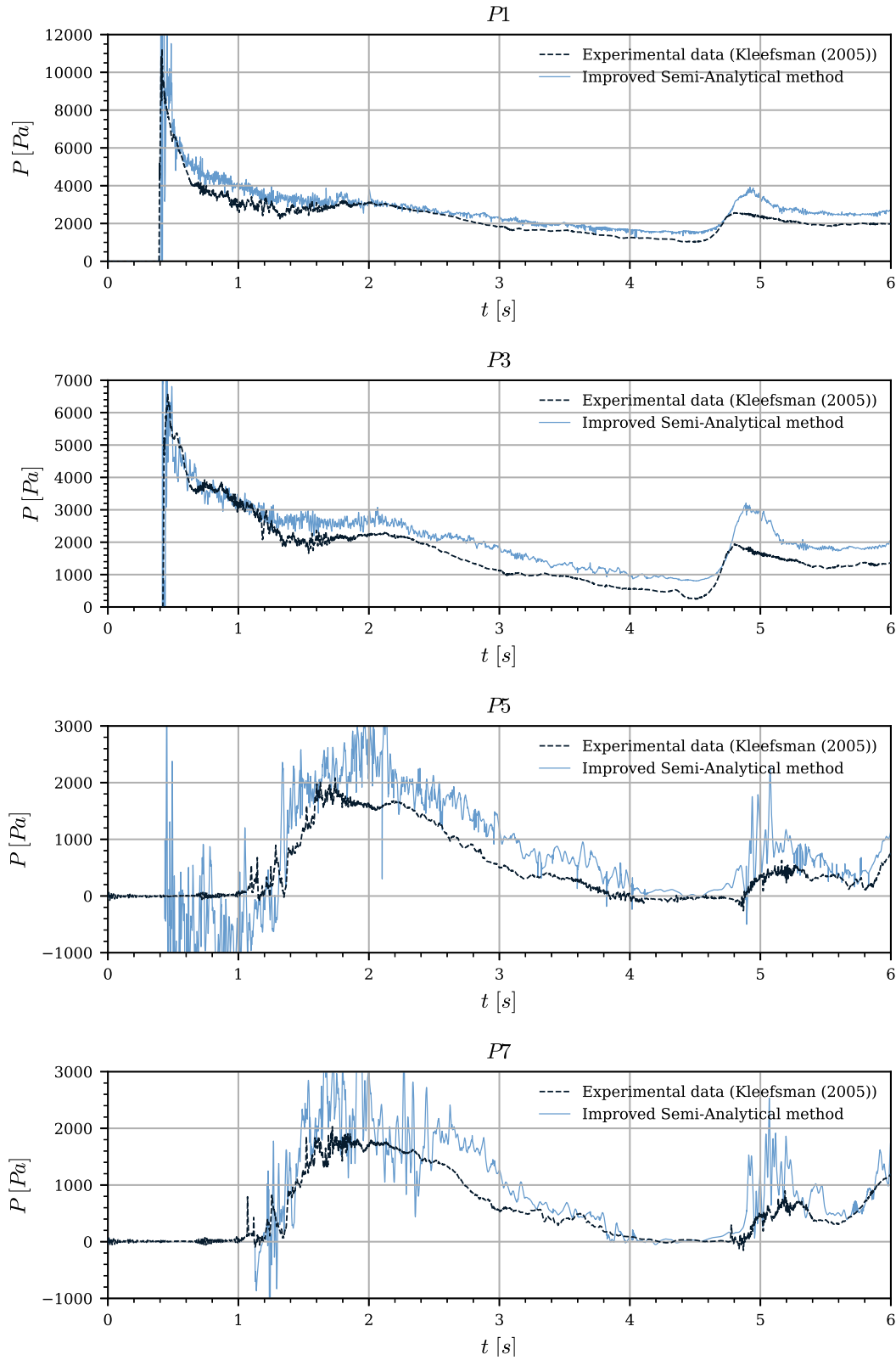


Figure 23: Time evolution of the pressure at given nodal locations based on the 3D dam break experiment by Kleefsman [81]. Plots are shown for four of the pressure nodes: $P1$, $P3$, $P5$, $P7$. The first two nodes are placed on the front of the obstacle where the dam break initially collides, considered upstream pressure nodes, while the latter two are positioned on top of the obstacle where the dam break later splashes on to after the first collision, considered downstream pressure nodes.

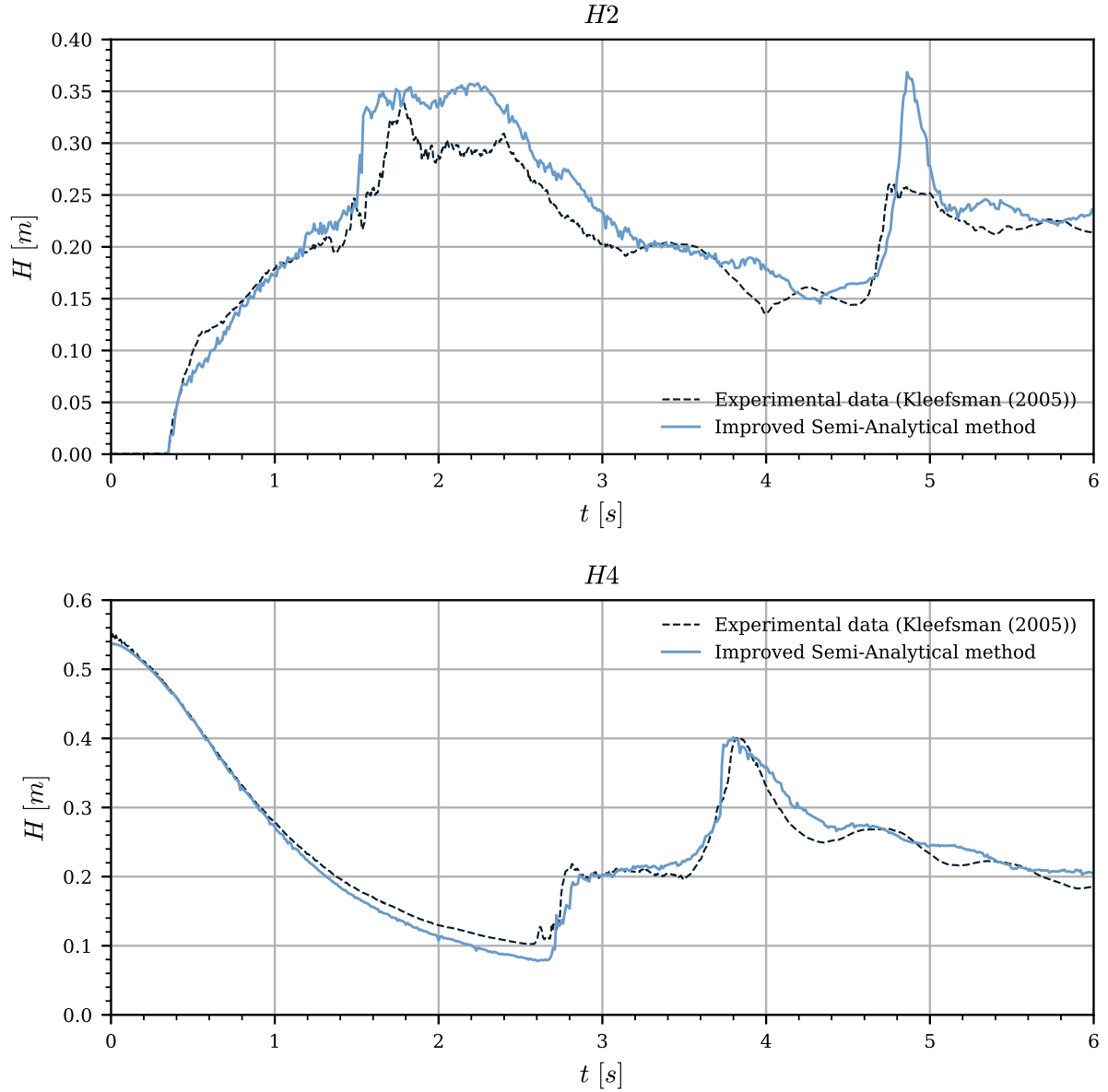


Figure 24: Time evolution of the free surface height at given locations based on the 3D dam break experiment by Kleefsman [81]. Plots are shown for two of the locations where free surface height is measured: $H2$, and $H4$. Both locations are upstream of the obstacle collision front, and $H4$ notably starts within the initialised fluid.

locations suffer from inconsistent particle supports, with fluid particles constantly passing at high speeds and being moved away from the node locations due to jets and crashing wave fronts. As previously discussed in Sec. 2.3, SPH is known to produce noisy and inaccurate pressure fields when particle supports are not regular. On the other hand, the upstream pressure nodes, $P1$ and $P3$, produce much smoother pressure fields and follow the experimental results more closely. In these locations fluid particles are more consistently located and are not subject to as many crashing wave fronts other than the initial impact of the water against the obstacle. Free surface height is generally well modelled throughout.

To further examine and assess the accuracy of this method, the results were also benchmarked against results that could be extracted from well-respected literature on other boundary conditions carrying out the same test simulation. Specifically, results from Mayrhofer et al.s unified semi-analytical boundary condition (USA-BC) extension to 3D [105], and a dynamic boundary particle method in English et al.s modified dynamic boundary condition (mDBC) [49] which is the same method currently used in the open-source code DualSPHysics. Data was extracted via plot digitisation from the published papers. Comparisons of results were partly limited as only certain markers were analysed in all three works compared. In this case, free surface heights at $H2$ and $H4$, and the downstream pressure node $P5$. All three boundary conditions were compared at roughly equivalent resolutions with a fluid particle number $n_p \in (5 \times 10^5, 5 \times 10^6)$. Results are presented in Figs 25 and 26.

It can be seen generally that the performance of the improved semi-analytical (iSA) boundary method here holds up well when compared with the literature. The comparable semi-analytical boundary method in the literature, Mayrhofer et al.’s USA-BCs [105], seems to perform poorly on the data points assessed here. Whilst there is interestingly almost no noise in its calculation of the pressure field, it fails to accurately capture the size of the pressure peaks and the following decay. Similarly, it fails to accurately capture the progress of the wave fronts in the free surface height. Meanwhile, the iSA method here and English et al.’s mDBC [49] seem to capture the overall flow dynamics of both pressure field and free surface height relatively well. The mDBC produces a notably much less noisy pressure field at $P5$ than the iSA method. Earlier comments suggested that the noisy pressure field may be due to the lack of particle support, however since this effect is not apparent in the literature results, it would suggest that the effect may be due to a different issue in this current implementation. It is noted that the mDBC does exhibit a noisy pressure field during the arrival of the first wave front, and later in the simulation after the first wave front has passed - a shared quality with the iSA method. In fact, after the first wave front has passed, the iSA method performs roughly up to par with the mDBC and in some periods, even slightly better. Though not presented here as data was not able to be collected for

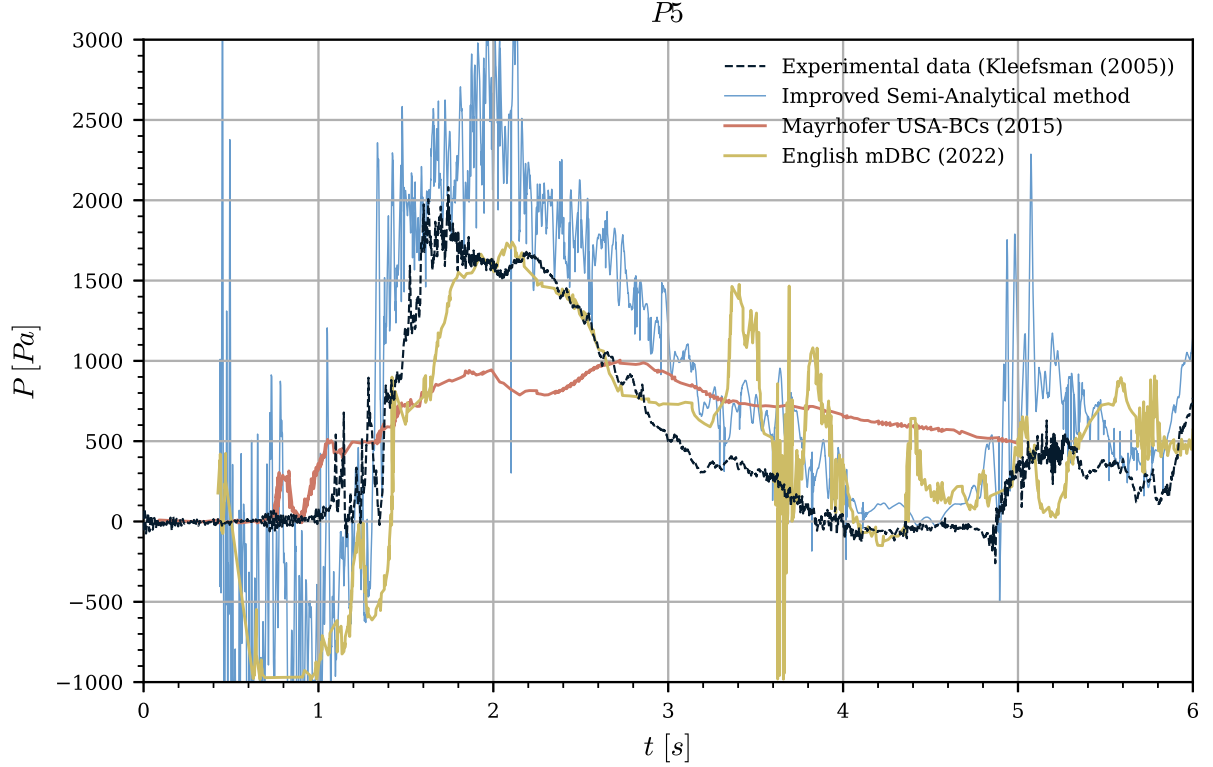


Figure 25: Time evolution of the pressure at $P5$ based on the 3D dam break experiment by Kleefsman [81]. Results are shown for experimental data alongside three different boundary method implementations: improved semi-analytical boundary methods from this work; Mayrhofer et al.'s unified semi-analytical boundary conditions in 3D [105]; and English et al.'s modified dynamic boundary condition [49]. Particle resolution is roughly the same for all three simulations with a fluid particle number $n_p \sim \mathcal{O}(10^6)$. Other simulation parameters roughly followed the test case guidance, though Mayrhofer et al.'s work used a numerical speed of sound $c_0 = 40$ m/s as opposed to the standard $c_0 = 60$ m/s.

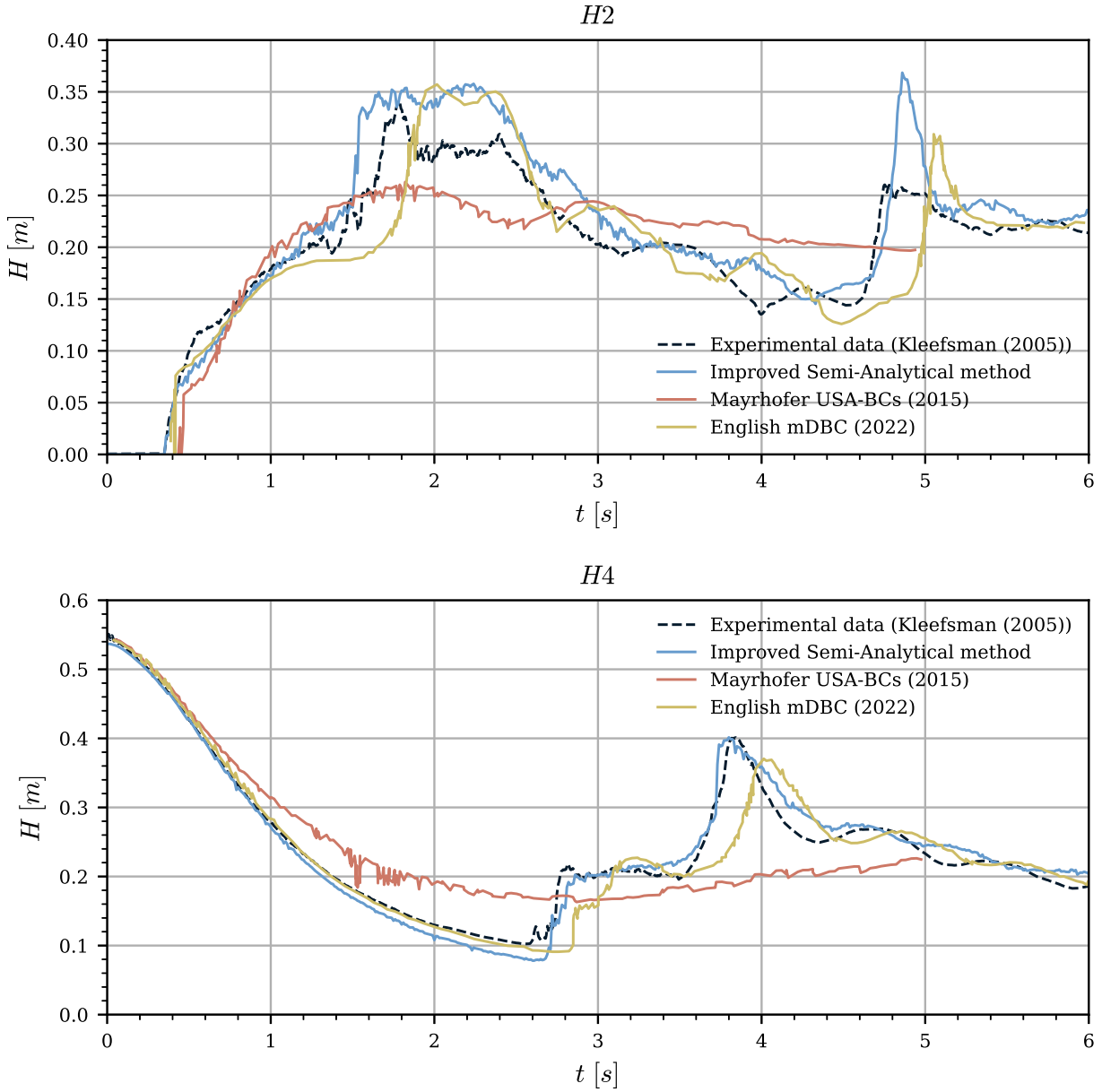


Figure 26: Time evolution of the free surface height at given locations based on the 3D dam break experiment by Kleefsman [81]. Plots are shown for two of the locations where free surface height is measured: $H2$, and $H4$. Results are shown for experimental data alongside three different boundary method implementations: improved semi-analytical boundary methods from this work; Mayrhofer et al.'s unified semi-analytical boundary conditions in 3D [105]; and English et al.'s modified dynamic boundary condition [49]. Particle resolution is roughly the same for all three simulations with a fluid particle number $n_p \sim \mathcal{O}(10^6)$. Other simulation parameters roughly followed the test case guidance, though Mayrhofer et al.'s work used a numerical speed of sound $c_0 = 40$ m/s as opposed to the standard $c_0 = 60$ m/s.

equivalent pressure nodes, both the iSA method and the mDBC appear to perform similarly on the upstream pressure nodes, exhibiting a relatively smooth pressure field and capturing the overall dynamics well.

Both the mDBC and iSA method capture the free surface height at both $H2$ and $H4$ well, with each other performing respectively better in different time periods of the flow.

Overall, the iSA method here is less accurate than English et al.'s mDBC when capturing the pressure field at $P5$, but much more accurate than Mayrhofer et al.'s USA-BCs as a comparable semi-analytical implementation. The main issue affecting the results is the noise in the pressure field, but otherwise the overall dynamics are well captured. The iSA method performs comparably to the mDBC for free surface heights, and even better in some periods, whilst again producing overall more accurate results than the USA-BCs. This suggests that the accuracy of this semi-analytical implementation in these types of 3D dam break simulations are up-to-par with well developed and modern methods in the literature, and perhaps even better than the semi-analytical alternatives.

It should be noted, that measuring the free surface height is not always trivial in SPH, especially when splashing or breaking of the free surface is involved. Care should be taken to ensure that the true free surface height is measured, but even this is sometimes ill-defined when jets and voids form within the flow. This may also explain some of the discrepancies with both experimental data and between different simulations.

Next, in order to study the convergence of the improved semi-analytical method and better understand how it scales with changing resolution, the same initial pressure and free surface height plots are shown for multiple resolutions: $\Delta x = H/11, H/22, H/55$ corresponding to roughly $n_p \sim 5,000, 50,000, 500,000$ particles respectively. These will be referred to as the low resolution, intermediate resolution, and high resolution simulations respectively. Results are presented in Figs 27 and 28.

Interestingly, across all pressure results, and in certain free surface height results, it does not appear that the method displays convergence at the studied particle resolutions. It appears that the lowest resolution $\Delta x = 0.05$ m is less accurate, as expected, but so too is the highest resolution $\Delta x = 0.01$ m, surprisingly.

First, considering the pressure fields. Both the higher and lower particle resolutions appear to produce noisier pressure fields than the intermediate particle resolution $\Delta_x = 0.025$ m at all four pressure node locations. The difference is particularly pronounced in the downstream pressure nodes $P5$ and $P7$. The increased noise in the higher particle resolution is mostly confined to high speed interaction regimes involving breaking waves, while during more settled flow periods, particularly on the upstream nodes $P1$ and $P3$, both the intermediate and highest resolutions display relatively low noise and accurate results.

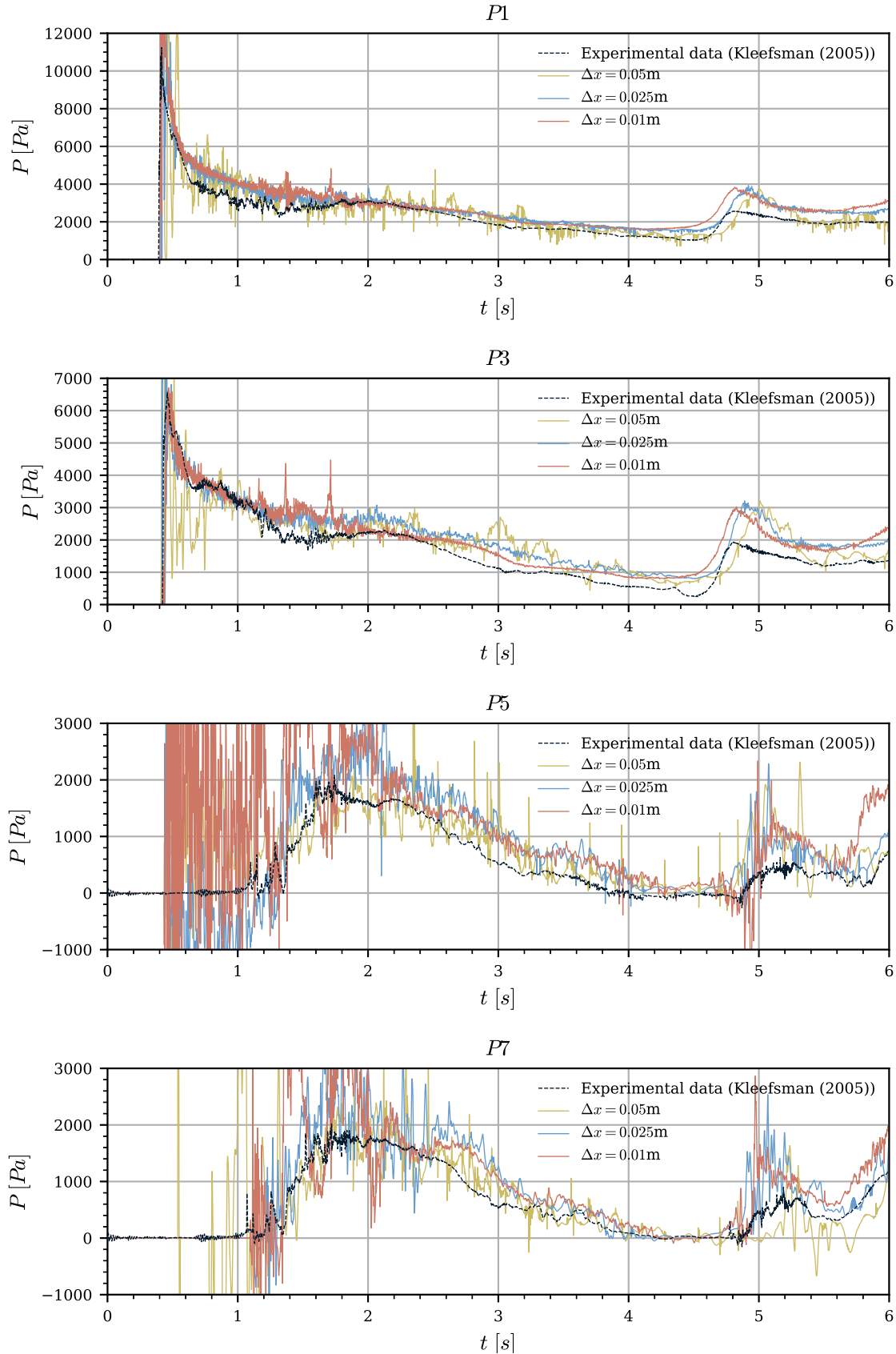


Figure 27: Time evolution of the pressure at given nodal locations based on the 3D dam break experiment by Kleefsman [81]. Results are repeated for three Δx resolutions, displaying how the result scales with an increasing number of particles.

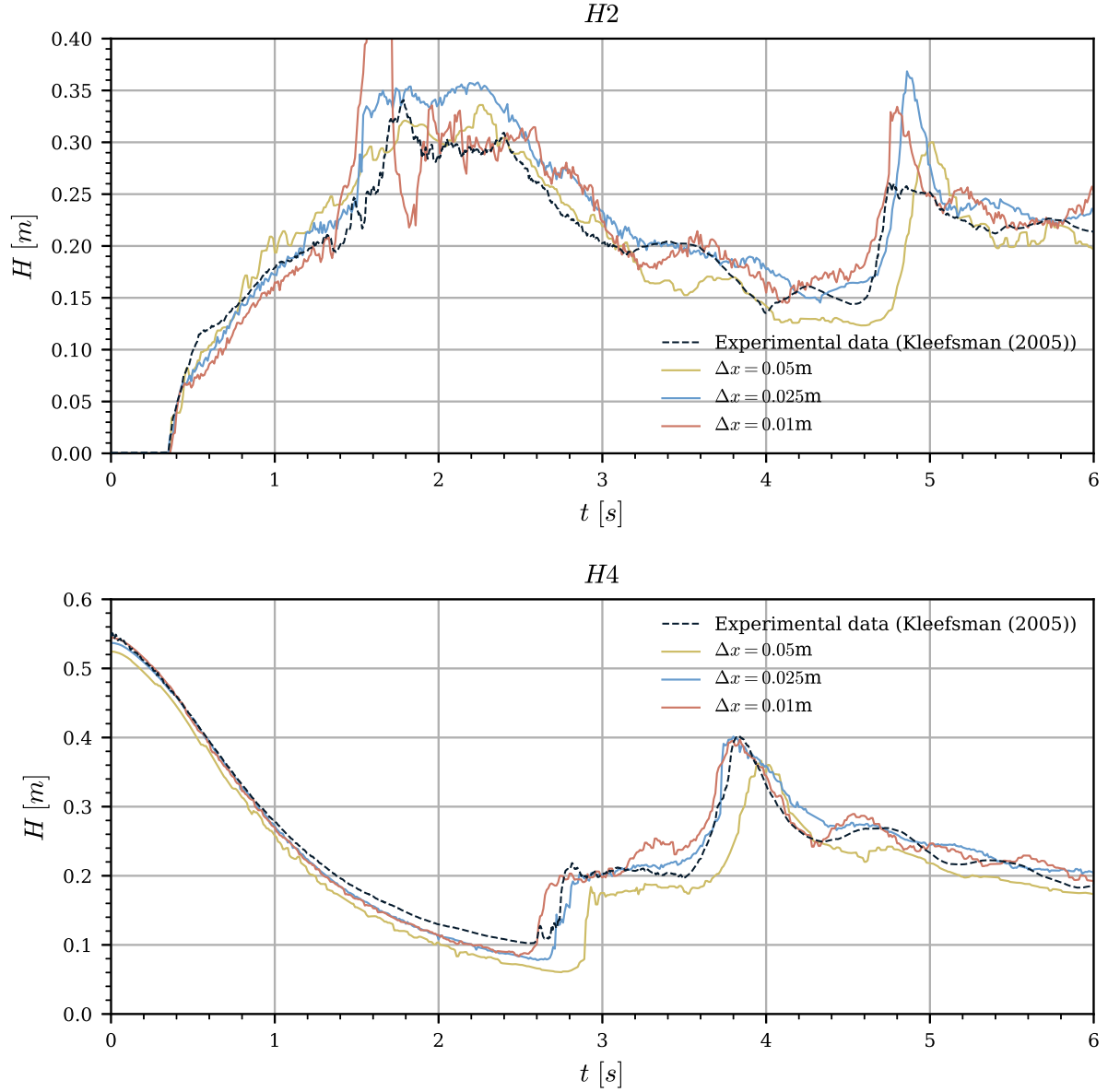


Figure 28: Time evolution of the height of the free surface at given locations based on the 3D dam break experiment by Kleefsman [81]. Results are repeated for three Δx resolutions, displaying how the result scales with an increasing number of particles.

The lowest resolution displays noise throughout and can be clearly marked as the least accurate resolution.

It was noted when viewing the simulation dynamics that the highest particle resolution displayed much greater formation of jets and greater forcings at the boundaries. As such, though the overall pressure field of the highest resolution was indeed the smoothest, the overall flow dynamics became much more noisy with greater wave crash heights and high velocity fluid jets near impact points. The intermediate and lowest resolutions on the other hand displayed similar and consistent flow dynamics in agreement with the experimental data. This also explains why the free surface heights in Fig. 28 are generally well in agreement and appear to display a slight convergence across particle resolutions, except for a specific point of the free surface height at $H2$; when the wave front first collides with the obstacle, the highest particle resolutions displays a brief strong divergence from the experimental data, briefly but largely overestimating and then underestimating the free surface height. This was observed as a high velocity fluid jet in the simulation upon first impact with the obstacle. These jets also occurred at other impact points throughout the back wall of the simulation at much lower strengths, but still generally enough to alter the local flow dynamics.

A potential reason for why the flow dynamics diverge at higher particle resolutions is due to inconsistencies in the boundary method being amplified due to the larger number of boundary interactions. It has already been noted that there can be large boundary inconsistencies with the method (Sec. 5.2) and it is possible that these exhibit some constant error which does not scale strongly with particle resolution. As such boundary calculation errors may become amplified as particle number increases. This is a notable effect, and should be investigated further in future work to develop this implementation.

It should be noted that in order to achieve the closest agreements to experimental data for dam break flows such as this one, it is generally understood that multi-phase simulations are required, as has been shown in the literature [110].

Ultimately, the simulations presented here only serve as a preliminary outlook into assessing the accuracy and feasibility of the improved semi-analytical method. For an SPH implementation and its associated boundary method, it is important that a variety of validation test cases are carried out. SPHERIC [149] offers a number of additional SPH validation test cases for different simulations regimes. Results should be compared not only to high quality experimental data, but also to high quality numerical data, such as those from well-established and validated mesh-based methods. Experimental data is often subject to repeatability issues, and Lobovsky et al. showed this specifically for the case of the 3D dam break [94]. Pressure readings in particular can have a high variance, and this should be considered when assessing the accuracy of a numerical method. Validations against analyti-

cally known solutions are also ideal, as these allow for more precise error tracking and more rigorous convergence studies.

6 Mobile boundaries in SPH

The problem of mobile boundaries in SPH requires careful consideration because not only is it an issue of fluid-solid coupling, but also solid-solid coupling, whereby boundaries are now free to move and interact with each other.

The fluid-solid regime of the problem is essentially just an extension of the fixed/wall boundaries problem previously discussed in Chapter 3. All the issues related to modelling wall boundaries are present in mobile boundary modelling, but further complications are presented. Particularly, ensuring that not only are the wall-to-fluid force calculations accurate, but also the fluid-to-wall force calculations. Forces on mobile boundaries must be accurately calculated and tracked, and then applied in such a way that the boundaries' motion can be accurately simulated. Further, it may be necessary to handle mobile boundaries which are not constrained by the resolution of the fluid particles, in order to allow for flexibility in the sizes of mobile solids modelled. Consideration should be given to mobile solids at both higher and lower resolutions than that required for the fluid domain.

Solid-solid coupling was briefly addressed in the introduction, and typically requires an additional numerical method to handle these interactions. Whilst SPH is a particle-based method and can in theory be used to handle solid interactions, methods which are specifically developed for granular materials and solid mechanics, such as the Discrete Element Method (DEM), are generally preferred.

In this work, DEM is used to handle solid-solid interactions and is coupled with SPH for the fluid-solid interactions.

In the following sections, an overview of the fluid-solid coupling process that was developed for this work, as well as the implementations of DEM for both large and small bodies, and their coupling to the thus far discussed SPH method, will be presented.

6.1 Fluid-Solid coupling

As mentioned, the basics of Fluid-Solid coupling has already been addressed in Chapter 3 on wall boundaries. Indeed, any of the discussed methods for modelling wall boundaries can also be used to model the interaction of mobile boundaries with the SPH fluid particles. One key exception to be considered, however, is when the boundaries are small enough that the discretisation required in a wall modelling method cannot be applied. This would be the case for any of the ghost particle methods, which are naturally limited by the size of fluid particles, and similarly, the current version of the semi-analytical boundaries which requires the surfaces of walls to be discretised with wall particles. In these cases, the limit to the mobile (or even fixed) boundary size would be on the order of the fluid particle resolution

$\mathcal{O}(dx)$. For ideal results, one would expect that boundaries which require discretisation should contain enough fluid particles to be accurately modelled. This would suggest a more effective minimum size limit of $\sim \mathcal{O}(10 \cdot dx)$. For fixed wall boundaries, this is not often considered an issue, as the fixed wall boundaries usually define the fluid domain and therefore are designed to meet the desired fluid resolution. Indeed, a highly resolved fixed wall boundary would necessarily imply a highly resolved fluid domain.

However, introducing mobile boundaries of a different phase allows the possibility of separate length scales within the simulation domain. For mobile bodies larger than the fluid length scale, it is essentially possible to fully resolve the flow structure in and around the bodies, and as such fully resolve all the interactions between the two phases. This is often preferred, as it allows interactions to be implicitly modelled from the governing equations with no additional closures required. This is the typical approach taken in SPH-DEM, whereby the DEM particles are used to model solids which are at length scales sufficiently larger than the fluid length scale. This is not always feasible however. For many applications, the dynamics of interest occur over widely varying length scales. Importantly, in the contextual case of coastal dynamics, one may be interested in simultaneously modelling the large length scale behaviour of storm or tsunami waves approaching a coastline, and at the same time considering the smaller scale dynamics involved in solid particles embedded in the coastline or seafloor. These particles such as pebbles, gravel, and sand can reach grain sizes as low as 0.1 mm [164] and even lower. Attempting to fully resolve the fluid flow in and around all these solid bodies would lead to excessively large resolution on the fluid domain and prohibitively computationally expensive simulations. Therefore, particularly when considering coastal dynamics, it can be valuable to admit mobile solid boundaries which are not constrained by the fluid particle resolution. This necessitates an alternative method of modelling granular solids which does not rely on any form of fluid particle discretisation.

It is often the case that the difference in length scale between the fluid and solid domains is so drastic that it is even unfeasible to model the solids as granular discrete particles, and they are instead modelled as a continuum, as a non-Newtonian fluid. This could be considered for fine sand particles which are fully saturated with water, where it is clearly computationally unfeasible to attempt to model the individual grains of sand. Using non-Newtonian fluids to represent granular material in two-phase simulations was briefly mentioned in Chapter 1, and indeed it is often the solid model of choice in SPH simulations [56, 146]. An advantage of modelling solids as a non-Newtonian fluid is that the SPH method no longer needs to be coupled with an additional numerical method to handle solid interactions. Instead, the simulation simply follows a two-phase model, and the existing framework for SPH can be extended to a non-Newtonian regime which has additional governing equations that affect the

density, velocity, and viscosity [56, 57]. Further, it is much less computationally expensive as it is possible to avoid the use of highly resolved discrete solid bodies, and the non-Newtonian phase can be cheaply solved within the main SPH fluid loops. It is not without its drawbacks however. Continuum models of solids are not always able to accurately model granular material in certain regimes, and information is lost on anisotropic behaviours that are relevant to scouring, damming, and granular segregation (bulk separation of the solid phase) [136], all of which are important in understanding coastal dynamics and the resulting fluid-solid interactions outlined in Chapter 1.

The work in this Chapter will focus on the mesoscale regime of fluid-solid interactions, above the length scale where solids are modelled as non-Newtonian fluids, but below the length scale where the inter-solid flow can be fully resolved. A relevant piece of literature for SPH at the fully-resolved fluid-solid length scale is by Bartzke et al. [15], who simulated flow past a sphere on a rough bed with very high resolution fluid, and found promising results in comparison to the literature.

Another issue in fluid-solid coupling in SPH methods, is the accurate calculations of fluid-to-wall forcings. For boundary modelling methods which involve some form of wall particle, fluid-to-wall forcings can be simply deduced as follows. Contributions from a wall particle to the local fluid particles' accelerations (from the momentum equation (27)) are tracked and summed, and then reflected to calculate an acceleration for the wall particle in question. The accelerations of wall particles are then appropriately applied to the body they are attached to in order to produce forces and torques due to the fluid interaction. Due to the antisymmetry of the standard momentum equation (27), this is equivalent to independently calculating the accelerations for wall particles by summing over local fluid particles. This is particularly relevant for the ghost particle methods discussed in Sec. 3.2. The semi-analytical boundary method discussed in Sec. 3.3 offers more ambiguities in generating fluid-to-wall forces. Though the method in the current presentation admits wall particles, the momentum equation under the semi-analytical framework is no longer antisymmetric, and so simply reflecting the wall-to-fluid contribution is no longer equivalent to independently calculating the wall particle accelerations due to local fluid particles. However, calculating accelerations for wall particles via a sum over the fluid is not necessarily a clearly defined behaviour. It would therefore seem preferable then, that when wall particle accelerations are required, their values should be taken as a reflection of fluid particle contributions in order to generate accelerations on the wall particles that at least conserve momentum. Further work may be done on this in order to identify the preferred approach.

Of course, in the instance where the boundaries are too small to be discretised with fluid particles as previously discussed, this method of calculating fluid-to-wall forcings cannot

work. For a non-Newtonian fluid based solid phase, the force interactions can be calculated as any SPH fluid interactions are done, just with additional constitutive and governing equations. For the mesoscale regime where discrete solid bodies are still resolved, but not the fluid within them, the forces between the fluid and the wall must be calculated differently. This will be the focus of “small-body DEM” (sDEM) in the following Sec. 6.3.

First, a brief introduction to how solid-solid interactions are handled via the DEM method will be presented. As the focus of this work was on fluid-solid coupling, this is kept brief. The coupling of SPH and DEM is generally well researched, as is the DEM method itself, and the literature contains a more in-depth discussion on the specifics of DEM [25, 165, 68].

6.2 Large-body DEM

The movement of a solid object is found by considering its interactions with both the fluid and solid bodies and calculating the total forces on the body due to these interactions. Typically, as discussed, solid objects are discretised with k wall particles with fixed relative positions. The force on each solid body $\mathbf{f}_k$ per unit mass (i.e. acceleration), can be broken down into body forces $m\mathbf{g}$, hydrodynamic fluid forcings $\mathbf{f}_{k,f}$, and rigid solid-solid contact forces $\mathbf{f}_{k,c}$. As previously discussed, the hydrodynamic fluid forces can be found by reflecting and summing the contributions of the wall particles to the fluid particles’ accelerations, which are found during the main loop of fluid particle interactions over the boundaries. The force on each solid body is then summed to find the total driving force on the object. The motion is then derived from the discretised form of Newton’s laws of rigid body dynamics:

$$M \frac{d\mathbf{V}}{dt} = \sum_k m_k \mathbf{f}_k = \sum_k \mathbf{f}_{k,f} + \mathbf{f}_{k,c} + m_k \mathbf{g} \quad (162a)$$

$$I \frac{d\mathbf{\Omega}}{dt} = \sum_k m_k (\mathbf{r}_k - \mathbf{R}_0) \times \mathbf{f}_k \quad (162b)$$

where M is the mass of the object, I the moment of inertia, $\mathbf{V}$ the velocity, $\mathbf{\Omega}$ the rotational velocity and $\mathbf{R}_0$ the centre of mass. Eq. (162a) solves the linear motion of the solid object, whilst Eq. (162b) is the rotational motion. These equations are then integrated in time to predict $\mathbf{V}$ and $\mathbf{\Omega}$ of the body, which allows for a full translational and rotational motion of the solid object as a whole. Whilst the fluid forces and fluid-solid interactions can be computed with the SPH method, the rigid solid-solid contact forces are often handled with the discrete element method (DEM). This leads to a coupled SPH-DEM numerical solution which can handle fluid-solid and solid-solid interactions.

The contact force between two solid bodies is modelled as a spring force and decomposed into a normal and tangential force, which are then further decomposed into a repulsion force due to solid deformation and a dissipation force. The deformation repulsion force is approximated by a particle overlap and represents solid collision, whilst the tangential force represents solid friction. This is known as the spring-dashpot model. A schematic of the DEM interaction can be seen in Fig. 29. A full study of the DEM force equations can be found in the literature [25, 165, 68].

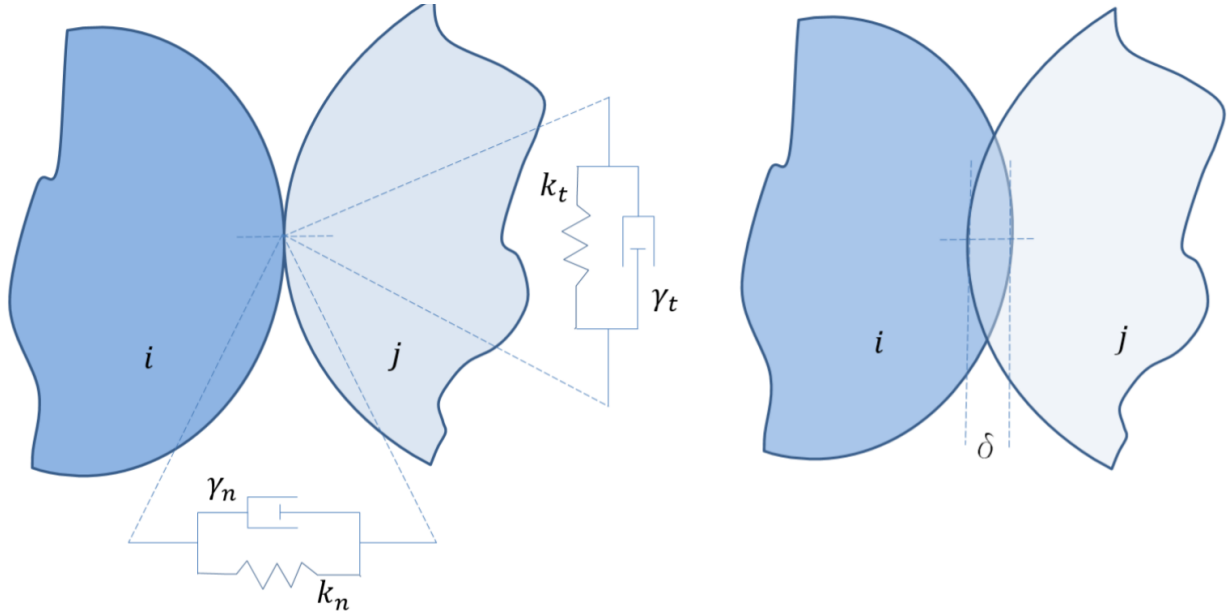


Figure 29: [24] Schematic of a DEM solid collision between two solid particles i and j . The left image highlights the modelling as a spring force in both the normal and tangential directions, where γ is the dissipative force constant and k the repulsion spring constant, with subscripts n and t referring to normal and tangential directions respectively. The force constants are calculated from input parameters for the solids considered. The right image highlights the particle overlap δ which is used to approximate the repulsion force.

The important parameters in the DEM equations are the Young's modulus, Poisson ratio, and the friction coefficient, which control the nature of the solid-solid interactions in the above spring-dashpot model for DEM.

The timestep for DEM calculations is based on the on the typical contact duration of a DEM interaction, which scales as the inverse of the Young's modulus. For this reason, the DEM timestep is often much smaller and more restrictive than the typical fluid timesteps considered in Sec. 2.7.1. DEM interactions can be subcycled on the fluid simulation. This involves holding the fluid and its contributions constant whilst the DEM interactions are calculated, since the DEM bodies may undergo many cycles for a single fluid timestep. SPH fluid variables are then only updated according to their own timestep as required.

6.3 Small-body DEM

The issue of small-body DEM then, is concerned with instances where DEM particles are beyond the fluid particle resolution and as such cannot be discretised with wall particles. This necessitates an alternative way for calculating the fluid-solid interactions which previously relied on the presence of wall particles. Solid-solid contact forces and body forces however, are agnostic to the presence of wall particles and can still be calculated as for regular large-body DEM (Sec. 6.2).

6.3.1 Fluid particle equations

The governing equations for the mesoscale fluid-solid model are given based on the volume-averaged Navier-Stokes equations coupled to additional closure equations for drag and lift [8, 136]. This may also be known as the two-phase mixture theory for sediment transport and granular flow [129, 65]. The key equations relevant to an SPH implementation will be presented, but not their derivations. It is noted that the volume-averaged Navier-Stokes equations attempt to model the fluid variables per unit volume, accounting for the presence of granular material. Unresolved fluctuations in the fluid variables necessitate additional closure equations for drag and lift for example. The governing equations largely follow the work of Robinson [136].

$$\phi(\mathbf{r}) = 1 - \int_{\Omega_s} W(\mathbf{r} - \mathbf{r}_s) dV_s \quad (163)$$

$$\frac{\partial(\phi\rho_f)}{\partial t} = -\nabla \cdot (\phi\rho_f \mathbf{u}) \quad (164)$$

$$\phi\rho_f \frac{d\mathbf{u}}{dt} = -\nabla P + \mu \nabla^2 \mathbf{u} - \mathbf{f}_s + \phi\rho_f \mathbf{g} \quad (165)$$

As can be seen by comparison to the original fluid dynamics governing equations (21), the presence of the granular solid introduces the term $\phi(\mathbf{r})$ which is the local fluid porosity or void fraction, and $\mathbf{f}_s$ which is the fluid-solid coupling term due to the interaction of SPH fluid particles and local solid bodies. The fluid density is given subscript f to clarify the distinction now possible between the densities of the fluid and solid phase. The porosity is calculated as 1 minus the normalised solid fraction $\phi_s = \int_{\Omega_s} W(\mathbf{r} - \mathbf{r}_s) dV_s$, equivalently $\phi = 1 - \phi_s$. The solid fraction is found by integrating the kernel function over the volume of local solid bodies, which naturally gives the relative volume of solid bodies within the local region. As with most SPH approximations, it is usually understood that the solid fraction integral is too complex to be calculated, and so a discrete sum approximation is used instead. This idea will be addressed in more detail in the following Sec. 6.3.4.

The porosity acts to scale the fluid density as it appears in the governing equations, and so it is often simpler to redefine the density used in SPH as $\rho^* = \phi\rho_f$, which allows us to reuse our original governing equations (21) with the addition of the fluid-solid coupling term $\mathbf{f}_s$. Considering the weakly-compressible approach again, the pressure equation is now given as

$$P(\rho) = \frac{\rho_0 c_0^2}{\gamma} \left(\left(\frac{\rho}{\phi\rho_0} \right)^\gamma - 1 \right), \quad (166)$$

where it is important to note that the equation of state is defined as an internal equation for the fluid, and so the reference fluid density must also be rescaled to account for the porosity. This form of the equation of state ensures that pressure only varies due to fluid fluctuations and not varying porosity.

Finally, it is necessary to derive an equation for the fluid-solid coupling term $\mathbf{f}_s$. This term is equivalent to the hydrodynamic forcings on solid bodies discussed in the previous section 6.2 - $\mathbf{f}_f$ in Eq. (162a). Of course, its presence must be reflected in the fluid equations, hence the $-$ sign in Eq. (165). For large bodies, this term was implicitly calculated in the boundary method, but in the case of small bodies the fluid-solid coupling term will be a resolved force explicitly calculated. This will be handled in the following section 6.3.2. For now, this term will be denoted $\mathbf{f}_{s,f}$, representing the fluid-solid coupling force at a small body s . The fluid-solid coupling term at a fluid particle j due to local small bodies is given

$$\mathbf{f}_{j,s} = \frac{1}{\sum_b V_b W_{sb}} \sum_s V_s \mathbf{f}_{s,f} W_{js} \quad (167)$$

Essentially in this equation, the fluid particle finds its coupling force by summing over the coupling forces at the surrounding small bodies. This must be appropriately scaled so that the force on the fluid exactly balances the force on the solid. Since the coupling force on the solid $\mathbf{f}_{s,f}$ is first calculated via a sum over the local fluid particles (here denoted by subscript b), the final term being returned to the fluid must be normalised by this sum - $\sum_b V_b W_{sb}$. For any solid body fully immersed in the fluid flow, this sum will likely be very close to 1.

The standard SPH discretisations and formulations presented in Chapter 2 can then be used to discretise the governing equations (164) (165) and model the fluid motion. First, however, it is necessary to provide a form for the term $\mathbf{f}_{s,f}$.

6.3.2 Solid body equations

The equations of motion for the solid DEM particles appear initially the same as the large-body DEM equations (162). However, the form of the hydrodynamic fluid forcings $\mathbf{f}_{f,k}$ now

have to be treated differently, since the standard approach of collating reflected solid-fluid contributions on wall particles cannot be used. The subscript k will hereon be dropped in favour of s , as the fluid forcings are no longer defined at wall particles, but rather at the center of solid bodies. For the purpose of fluid-solid interactions, the solid bodies can be considered as SPH fluid particles which transmit and receive forcings differently, but nonetheless act as an integration node where field variables can be defined and evaluated. In general, all fluid-solid interaction forces will arise due to the fluid stresses, which are either pressure or viscosity interactions. However, in the case where the inter-particle fluid cannot be fully resolved, additional equations which parameterise the effects from the unresolved flow field must be used. This means that fluid-solid forcings are broken down into the standard resolved fluid forces, and additional inter-particle forces $\mathbf{f}_{IP}$ which are usually a function of the porosity, as well as other flow variables. A standard form can be written as

$$\mathbf{f}_{s,f} = V_s(-\nabla P + \mu \nabla^2 \mathbf{u})_i + \mathbf{f}_{IP} \quad (168)$$

and the interparticle forcings can usually be broken down into the following form

$$\mathbf{f}_{IP} = \mathbf{f}_D + \mathbf{f}_L + \mathbf{f}_B + \mathbf{f}_{AM} \quad (169)$$

Here, $\mathbf{f}_D$ refers to the drag force due to viscous effects, $\mathbf{f}_L$ refers to the lift force due to shear flow or rotating particles, and $\mathbf{f}_B$ and $\mathbf{f}_{AM}$ refer to the Basset force and added-mass force respectively, arising from viscous effects and unsteady flow [172]. Similarity to the Basset-Boussinesq-Oseen equation is noted [148]. The lift force can be broken down into an inertial component arising from a perpendicular velocity gradient, and a component due to the rotational motion of the solid body, known as the Magnus lift force. In this work, rotational velocities of the particles were found to be very low such that the inertial shear term was assumed dominant. The buoyancy force is implicit in the pressure gradient term from fluid stresses in Eq. (168), indeed because the buoyancy force is due to the hydrostatic pressure gradient.

To evaluate the contribution of fluid stresses at the position of solid body s , $(-\nabla P + \mu \nabla^2 \mathbf{u})_s$ in Eq. (168), a summation is completed from the solid bodies over the local fluid particles

$$(-\nabla P + \mu \nabla^2 \mathbf{u})_s = \frac{1}{\sum_f V_f W_{sf}} \sum_f V_f \theta_f W_{sf} \quad (170)$$

where

$$\theta_f = -\sum_j \rho_j V_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla W_{ij} \mathbf{e}_{ij} + 2\nu \sum_j V_j \frac{\mathbf{u}_{ij}}{|r_{ij}|} \nabla W_{ij} \quad (171)$$

is just the fluid stress at fluid particle f , calculated equivalently as from the fluid governing equations (27) and of course suitably adjusted for any additions or changes to the pressure gradient or viscosity formulation. It is noted here that this effectively requires two stages of summations, one to first calculate the fluid stresses at each fluid particle θ_f , and then another to extrapolate these fluid stresses to the solid body positions.

The sum in Eq. (170) has also been appropriately normalised by a Shepard Filter style sum as in Eq. (29).

Next it is necessary to calculate the inter-particle unresolved forcings, $\mathbf{f}_{IP}$. Most SPH implementations which implement a small-DEM model do so only parameterising the drag force [136, 65, 68]. This is primarily because it is well understood that the drag force and pressure gradient (equivalently buoyancy) are often the dominant hydrodynamic forcings on suspended particles [16], and it is thus often the case that the drag force and fluid stresses alone are enough to encapsulate the majority of the fluid-solid interactions. That being said, the role of the additional lift, Basset, and added-mass forces should not always be so simply ignored. For the process of saltation, whereby solid particles are entrained into the fluid flow and bounced along the channel bed in the streamwise direction, it has been shown that in certain regimes, not only do the shear lift and Basset force play a role in the motion of particles, but in some instances they are even necessary to observe accurate saltation behaviour as found in experimental results [5]. In this work, the focus was given to inclusion of the lift force specifically. The use of the drag force in mesoscale SPH-DEM is well studied, and the Basset force contains potentially difficult terms to evaluate, relying on the history of particle paths. As such, the aim was to observe whether inclusion of the lift force was feasible, and more importantly whether the regimes suggested in the literature suggesting the importance of the lift force in entrainment and saltation, could be replicated with a mesoscale SPH-DEM model. A diagram depicting saltation of solid particles can be seen in Fig. 30.

The inter-particle forcings included in this model were the Gidaspow drag model [61], and Saffman shear lift force [137]. The Gidaspow drag model is given as

$$\mathbf{f}_D = m_s \frac{\mu_f}{\rho_g d_s^2} (\mathbf{u} - \mathbf{u}_s) D \quad (172)$$

with

$$D = \begin{cases} 18\phi^{-3.65} (1 + 0.15 (\phi Re_p)^{0.687}) & \text{for } \phi_s \leq 0.2 \\ 150 \frac{\phi_s}{\phi} + 1.75 \frac{Re_p}{\phi} & \text{for } \phi_s > 0.2 \end{cases} \quad (173)$$

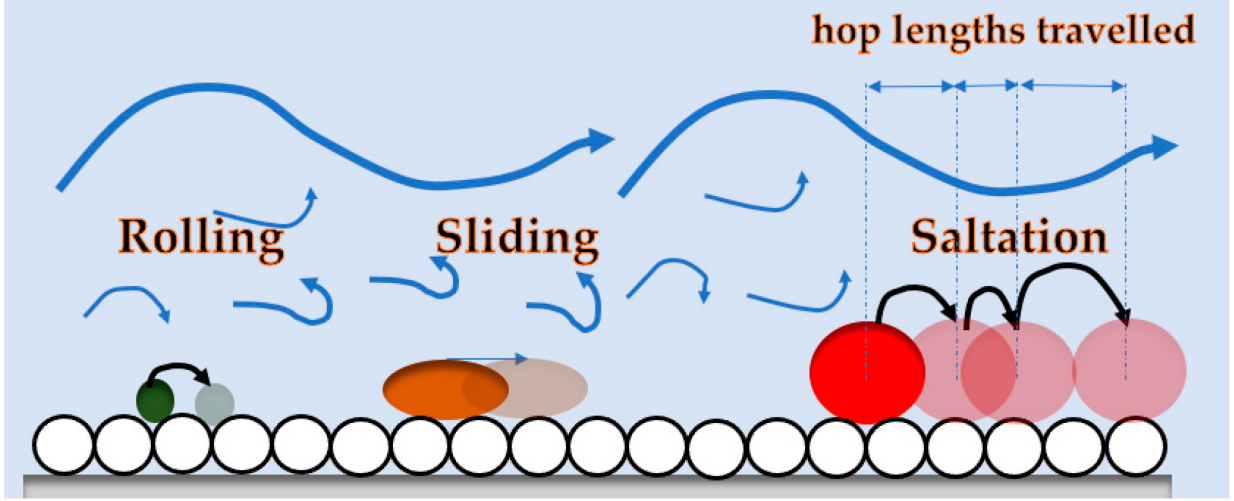


Figure 30: [4] Schematic demonstrating the main modes of bed load transport by rolling, sliding, and saltation, shown in green, orange, and red colors, respectively. A saltating particle often exhibits complex three-dimensional movement, but is particularly characterised by rising into the flow before falling again and then repeating this behaviour.

and the solid particle Reynold's number is given by

$$\text{Re}_p = \frac{\rho_f d_s |\mathbf{u}_s - \mathbf{u}|}{\mu_f} \quad (174)$$

Subscripts with f denote fluid variables, while the s subscript denotes solid variables. d_s is the diameter of the solid body. $\mathbf{u}_s$ is the velocity of the solid body, and $\mathbf{u}$ is the the fluid velocity extrapolated to the position of the solid body. This usually is calculated a via an initial sum over the local fluid particles $\mathbf{u} = \sum_f V_f \mathbf{u}_f W_{sf}$.

The Saffman shear lift force is given as

$$\mathbf{F}_L = C_L \frac{\rho_f \pi}{8} d_s^3 [(\mathbf{u} - \mathbf{u}_s) \times \boldsymbol{\omega}] \quad (175)$$

where C_L is the lift force coefficient, typically taken as 6.46, but here left open to user input as it is understood that the lift parameterisation may need adjusting to match validation experiments. $\boldsymbol{\omega}$ is the fluid flow vorticity at the position of the solid body. This is found via a two stage sum in the pre-calculations, where the vorticity is first found at each fluid particle $\boldsymbol{\omega}_i = -\sum_j V_j (\mathbf{u}_j - \mathbf{u}_i) \times \nabla_i W_{ij} \mathbf{e}_{ij}$, and then the vorticity at the solid body position is extrapolated from the fluid particles as $\boldsymbol{\omega} = \frac{1}{\sum_f V_f W_{sf}} \sum_f V_f \boldsymbol{\omega}_i W_{sf}$. The vorticity operator used is extrapolated from derivative forms in Eq. (19).

It is noted here that the inclusion of small solid bodies in the current formulation necessitates flow variables to be defined at the solid body positions. This requires previous

summations to be calculated to extrapolate said variables to the solid body positions before they can be used in the force equations (168). Some of these flow variables are even secondary variables, such as the fluid stresses and the vorticity, requiring a further additional loop to calculate these variables at each fluid particle before they can be summed. This is already considered in the typical SPH implementation loop as in Algorithm 2.7.3 in the *PreSolution Calculations*, but it should be noted as an extra consideration. The inclusion of DEM calculations in an SPH code is generally understood to make the implementation more cumbersome and greatly reduce simulation speeds.

6.3.3 Inclusion of a lift-force for solid particle saltation mechanics

A simple test-case experiment was simulated to assess whether the Saffman lift force as in Eq. (175) could play a significant role in saltation behaviour of solid particles within certain fluid-solid regimes.

The simulation set-up is presented in Fig. 31, whilst the results tracking specific bodies and their vertical motion are presented in Fig. 32.

Fig. 32 demonstrates how the presence of the Saffman lift force indeed did generate more pronounced saltation behaviour for the DEM bodies in the channel bed. This is in agreement with some of the results presented by Ali [5], which suggested that at larger particle diameters of $d_s = 30$ mm, comparable to gravel, the lift force was required to fully capture the trajectory of saltating particles. Without the lift force modelled, particles' trajectories were underestimated. It is noted that this is specific to this range of solid particle size, and the result should not necessarily be extrapolated to even smaller solid bodies on the scale of sand.

6.3.4 Semi-analytical calculations of porosity

The work in this section is in part extracted from a paper published during the course of this PhD [85].

The method proposed in Sec. 4.3 was implemented in the current SPH code for the purpose of evaluating the solid fraction (or equivalently, porosity) in the presence of many small DEM spherical bodies. As previously mentioned, when solids which are smaller than the fluid particles are handled (and even in other cases), it is often necessary to accurately know the local solid fraction. This is necessary both for the explicit fluid-solid coupling equations in Sec. 6.3.2 and the calculation of superficial fluid density used in the governing equations (164) (165). The solid fraction at a given fluid position $\mathbf{r}_i$ is rigorously defined as

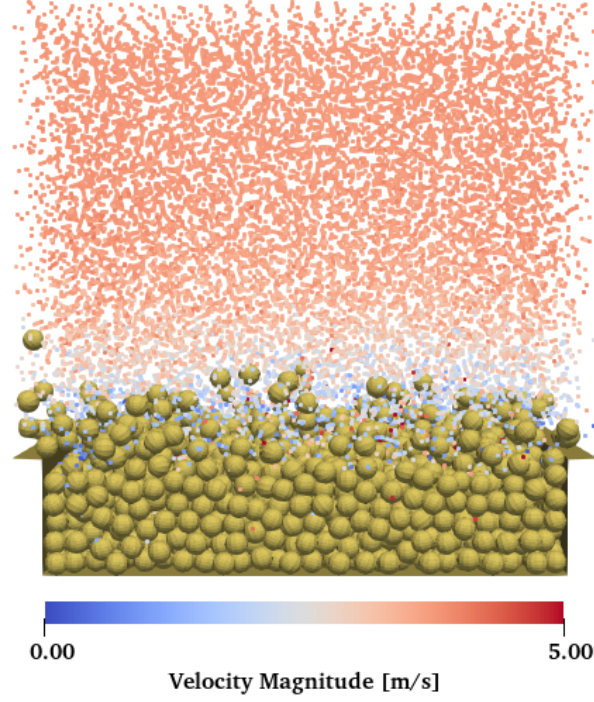


Figure 31: Simulation setup for periodic channel flow over a bed of mobile DEM bodies. Solid DEM bodies ranging from radii of $r = 10$ mm to $r = 15$ mm, with respective densities ranging from $\rho_s = 1700 \text{ kgm}^{-3}$ to $\rho_s = 2700 \text{ kgm}^{-3}$ were initialised into the basin of a channel of water with dimensions of $(1.0, 0.5, 0.5)$ metres. The solid parameters of the DEM bodies were taken as typical values for a gravel/rock-like material. The fluid had a resolution of $\Delta x = 0.02$ m, giving DEM bodies a size in the range of 50% – 70% the fluid particle size. The channel was periodic in the x and y direction, and a periodic pressure gradient was then applied in the x direction to generate fluid velocity. This was maintained until a maximum velocity in the x direction of $v_x = 5 \text{ ms}^{-1}$ was reached. A selection of 7 DEM bodies were then selected and their positions tracked over time.

$$\phi_{i,s} = \sum_{j \in \mathcal{S}} \int_{\Omega \cap \Omega'_j} W_{ij} \, d\mathbf{r}'_j \equiv -\gamma^- \quad (176)$$

where j runs over all solid particles within the SPH particle support $\mathcal{S}$, and Ω'_j is the volume of the solid particle j . The formula thus represents the integration of W centered at particle i over the volumes of all the neighbouring particles within the support. γ^- is defined from Sec. 4.3.

As can be seen, this definition is easily calculated from γ_j^- , and so naturally lends itself to the calculations presented in Sec. 4.3. This suggests a “semi-analytical” style calculation for porosity. Notably, this calculation would be exact. The form most commonly used in

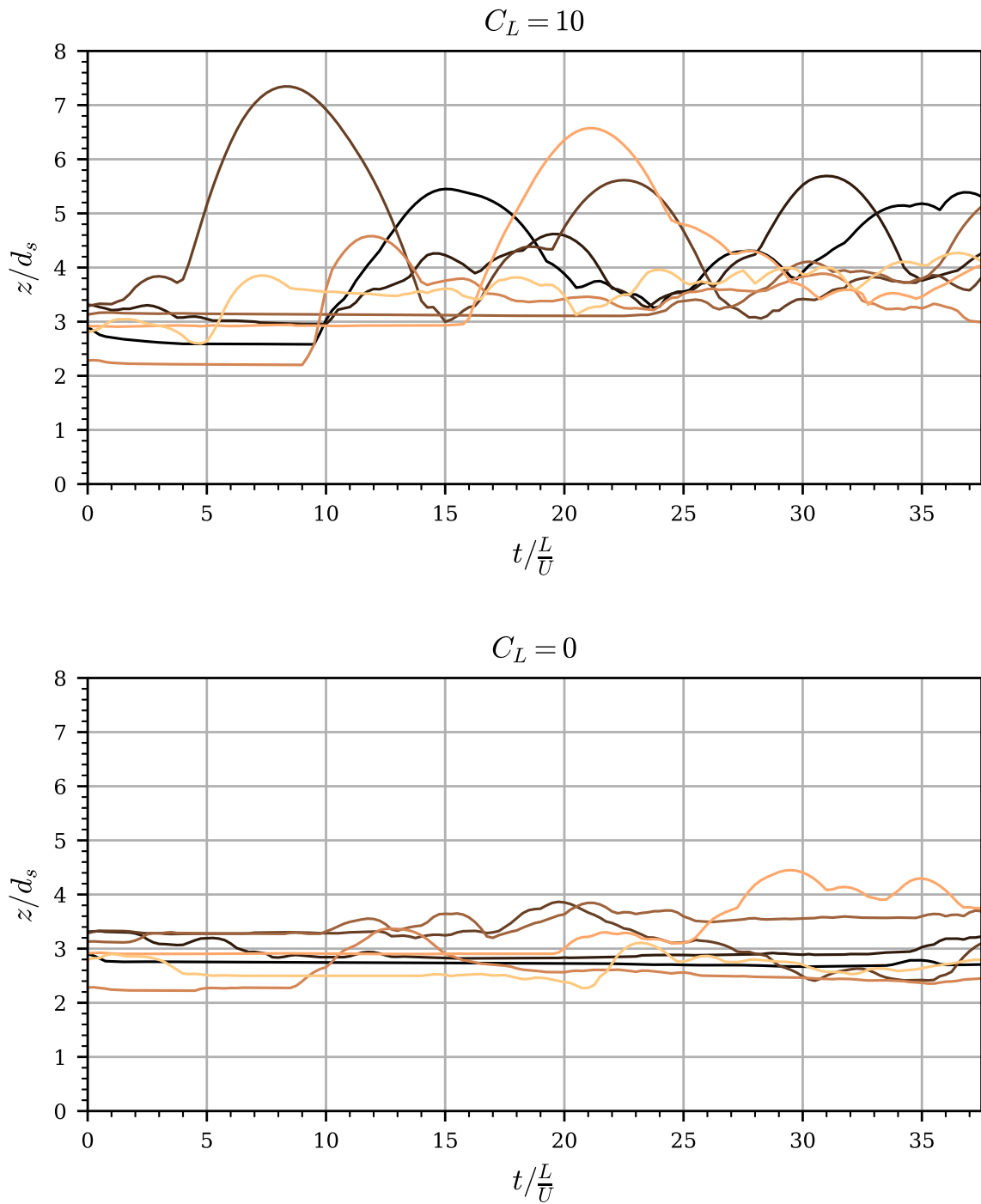


Figure 32: Plots of non-dimensional vertical positions vs non-dimensional time for the 7 DEM bodies selected from Fig. 31. This illustrates the vertical trajectories of the mobile DEM bodies during periodic channel flow both with and without lift effects. The first simulation in the upper graph was run with $C_L = 10$ and then repeated for the lower graph but with $C_L = 0$, highlighting the difference that the presence of the lift force in Eq. (175) plays in determining particle trajectories in channel flow over a mobile bed.

literature, however, is the SPH approximation of Eq. (176), given by

$$\phi_{i,s} = \sum_{j \in \mathcal{S}} W_{ij} V_j \quad (177)$$

where j again runs over all solid particles within the SPH particle support, $V_j = \int_{\Omega'_j}$ is the volume of spherical solid j .

The discrete estimate (177) works effectively as long as the solid particles are small enough (i.e. the discretization is fine enough), but starts to break down as the solid particle size approaches that of the fluid particle. It is commonly taken that in an SPH-DEM method the fluid particle smoothing length h should be sufficiently larger than the DEM particle diameter d_s , with $h \geq 2d_p$ suggested as the limit for accurate results [136]. Intuitively, Eq. (177) is equivalent to using the (non-composite) midpoint rule, which is expected to be generally accurate only if the integrand W is approximately constant over the domain of integration.

In order to demonstrate the aforementioned limitation and its mitigation via the method proposed, an illustrative test problem was studied. The results can be seen in Fig. 33. In this test, a three-dimensional body of fluid is initially at rest. Next, a layer of spherical DEM bodies denser than the fluid is grown in the following way. A flat square lattice is generated at a fixed height above the bottom of the fluid domain, and at each lattice point a DEM body with zero radius/volume is introduced. Due to the trivial size of the solid bodies, introducing them in this way has no practical effect on the dynamics and does not result in non-physical discontinuities. Next, the radius of the DEM bodies is gradually increased while their position is kept fixed. The growing process is stopped after the spheres reach a pre-specified size, and at this point the layer of grown spheres is released and starts settling under gravity. After a specified time interval (such that the released layer of spheres clears their original positions) a new layer is introduced, grown and released in the same fashion.

Initially, the DEM bodies are grown just above the body of fluid and gently splash onto the surface after being released. However, each next layer of solids increases the height of the free fluid surface and eventually the spheres are grown inside the fluid body. The initial zero size of the DEM bodies avoids non-physical behavior; the solids exert a force on the surrounding fluid as they grow from zero size and make it adjust itself appropriately in a smooth fashion. The main focus in this test is on the stability of the simulation owing to the accuracy of the porosity calculation.

As can be seen in Fig. 33, fully-grown body radii close to that of the fluid resolution (in this case 90%), cause a breakdown in the stability of the simulation for the discrete sum method, with fluid particles being unphysically thrown from the box at high speeds. This

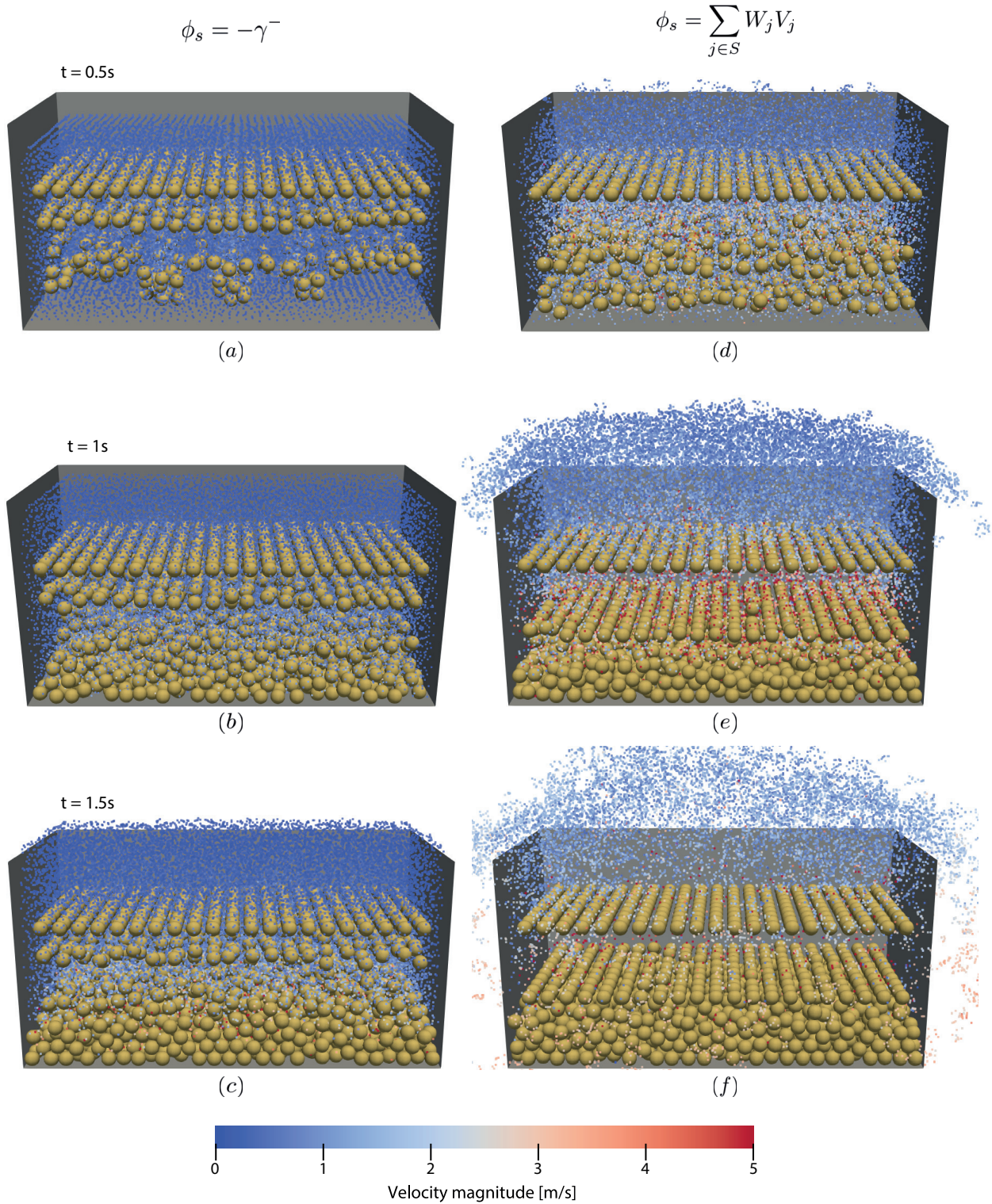


Figure 33: Visualisation of the SPH-DEM test problem. Small DEM bodies are gradually released in layers at a fixed height into a box of fluid of dimensions $(1.0, 0.5, 0.5)$. The fluid has a particle spacing $\Delta x = 0.02$ totalling 23520 particles. The bodies have a radius $r_s = 0.018$ - 90% the size of the fluid particles - and are regularly spaced in layers of 210. The layers are released every $0.16s$ until a final time of $1.5s$, so the three rows of plots correspond to the release times of the third, sixth and ninth layers. The left column of images ($a - c$) are implemented using a porosity calculation according to the method presented in this paper, whilst the right column ($d - f$) presents the typical SPH-DEM implementation of porosity (177).

is owing to the estimation of the porosity in equation (177), which largely overestimates the porosity when the body radii are not sufficiently small. An overestimated porosity leads to unphysically large forces and drops in fluid density. However, the current proposed semi-analytical method behaves stably at all times and produces physically consistent and accurate results for relative fluid/solid particle sizes closer to 1. The fluid stays stable as the solid bodies are dropped in and settle, and the total volume of the system gradually expands as is expected. The solid bodies are able to settle at the bottom of the fluid where porosity will be very high without producing erroneous forces and spurious velocities (cf around the second layer from the top in Fig. 33e). Note that with the semi-analytical method the distance between the top and the next layers is approximately constant in all three frames (Fig. 33a – c) while with the discrete approximation it is not (Fig. 33d – f). The latter is incorrect and is caused by the fluid being thrown from the box due to aforementioned numerical artifacts (hence the solids practically fall under gravity in near-vacuum).

The proposed method of calculating porosity via γ^- was then benchmarked against the usual method of a discrete sum over solid volumes in terms of computational cost. The aim was to assess whether the use of the method in Sec. 4.3 for calculating γ^- was significantly more expensive than the typical discrete sum as in Eq. (177). Initially, small scale simulations ran on a single node on a local machine were evaluated. These were designed to involve minimal computer communications and other overhead, such that the fluid-solid particle interactions dominated the computational expense. These simulations involved dropping small solid DEM bodies into a box of stationary fluid, calculated once with the semi-analytical method for porosity, and once with the discrete sum method. Generally, due to the size of the DEM bodies, the fluid dynamics were not appreciably affected, and the results with both methods appeared effectively identical. Average times per iteration were recorded. In these cases, it was found that the semi-analytical method for calculating porosity was $\sim 25\%$ slower, with a standard deviation of $\sim 10\%$. This result suggests that the porosity calculation from γ^- is roughly $\sim 25\%$ slower than the equivalent discrete sum calculation. This is perhaps to be expected - the basic discrete sum method contains just a single kernel evaluation and multiplication, while the proposed calculations in the semi-analytical method are much more involved - there are at least two *if*-statements and many more multiplications and divides. Whilst some of these processes can be optimised via modern compiler optimisations, nonetheless the semi-analytical calculation does carry noticeable additional computational expense compared to a standard discrete sum.

However, it is not always clearly the case that computation times are dominated by the fluid-solid interaction calculations. Rather, for large scale simulations on multiple nodes and with multiple computer architecture interaction points, there is a lot of overhead attached

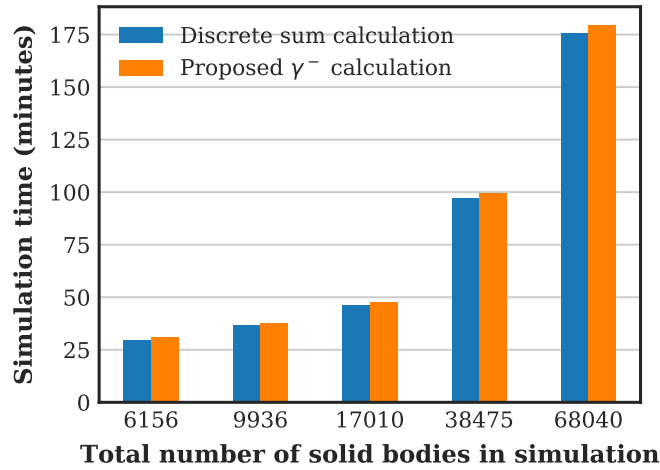


Figure 34: The total simulation times for two different methods of calculating porosity/solid fraction. Simulations were run parallelly on 48 cores of Intel(R) Xeon(R) Gold 5118 CPU @2.30GHz over 2 computational nodes. Simulation times were averaged over three independent runs. The number of solid bodies in the system are increased along the x-axis, which should present the opportunity for fluid-solid interaction calculations to dominate more of the computation time, even with much higher computational communication and memory overheads. At most, the number of solid bodies totals around 3/4 of the total system particles. The simulation is set up such that the number of bodies does not appreciably affect the dynamics. The standard deviation of the simulation times was less than 1% in all cases.

to communications and memory handling. In these instances, the fluid-solid particle interactions assume a smaller proportion of the total computational expense. This is more relevant and more realistic for high resolution simulations with a large number of fluid and solid particles. To this end, the simulations were repeated but at much large scales and with a larger number of computational nodes. The number of solid bodies and their diameters were varied for successive tests without affecting the overall dynamics; the number of solids was increased as their diameter was decreased and vice versa. This ensured that all simulations were stable and exhibited similar low velocity fluid dynamics whilst containing progressively more solid bodies. Additionally, investigating different solid diameters takes into account a range of relative fluid/solid particle sizes and changes the relative proportion of computational time spent on fluid-solid calculations throughout simulations. Results comparing the speeds of each method are presented in Fig. 34.

Fig. 34 shows that for large systems with a large number of fluid particles and inter-node communications, there is no appreciable difference in computation time presented by the proposed method for calculating porosity. This perhaps appears contradictory to our previous results suggesting $\sim 25\%$ greater computational expense for the semi-analytical calculation of porosity. This is likely because higher resolution simulations involve greater computer communications and memory handling, and so the fluid-solid particle interactions account

for a much smaller proportion of the total computational expense. This remains the case even when the number of solid bodies tested totals roughly 3/4 of the total system particles.

The conclusion therefore is that the relative additional expense of the semi-analytical calculation for porosity depends on the simulation set-up. For very large systems where the fluid-solid interactions are a small percentage of total computational expense, it is obvious that the semi-analytical calculation remains computationally feasible. However, for systems with extremely large numbers of small bodies, the discrete approximation can remain accurate and potentially faster. Of course, if the solid bodies need to be modelled at the mesoscale such that they approach the fluid resolution, then the semi-analytical exact approach should always be used to guarantee stability. It is trivial to create a pre-runtime switch between the two methods, allowing flexibility for the user.

7 Conclusions & Future work

The primary aim of this thesis was to study the smoothed particle hydrodynamics (SPH) numerical method with an aim to improve its viability for the use of modelling coastal dynamics. Whilst this aim is one already addressed in the literature, this thesis intended to focus on improving viability of the method at lower resolutions and cheaper computational cost, such that accessibility was improved at all levels. To this end, there was a concerted effort to maintain a fully explicit numerical scheme and achieve improvements in accuracy even at low resolutions. The hope is that all implementations are clearly explained and presented such that they could easily be implemented in an existing SPH code.

First, the SPH method was presented in full, with a focus on accuracy of the method, formulations of derivatives, and the most useful and easily implemented techniques for improving accuracy. Many of these methods are all well validated in the literature, and so the focus was rather on providing justification for methods that are often assumed to be standard in SPH. Some of the less discussed nuances were also explained, such as initialisation of an appropriate hydrostatic pressure field. Next, this discussion was extended to wall boundaries in SPH. Again, as many of these methods are already well studied and validated, the aim was to discuss and justify certain nuances of implementation, such as the spacing of wall particles and their pressure assignment.

One of the standard boundary methods discussed in Chapter 3, was the semi-analytical boundary method. This is a boundary method within SPH which aims to model the presence of boundaries without additional extraneous forcings or ghost particles as other boundary methods do. Instead, the appropriate forces and adjustments to the governing equations are generated implicitly, from a more complete derivation of the SPH operators. Due to the way this boundary method arises, it is often very desirable and can offer improved accuracies for certain problem types. Despite this, the method remains less popular in the literature, due to its complexity and the difficulty involved in accurately and efficiently calculating the highly irregular normalisation terms - γ and $\nabla\gamma$ - for all types of boundaries. The main work in this thesis focused on developing improvements to the semi-analytical boundary method, with the main achievement being a fully analytical solution to the normalisation terms, alongside an easily portable implementation algorithm. Previous literature had attempted this problem, and whilst good developments were made using a geometric formulation of the problem, all the solutions involved very convoluted formulae and additional domain decomposition processes, whilst still being limited in scope. This work, alongside the papers published, is

the first to present a fully general and analytical solution to the calculation of normalisation terms, requiring only the basic boundary definition variables - vertices and normal. This was done by utilising a carefully constructed geometrical formulation of the problem which decomposes simply into closed-form analytical terms, and more complex straight line integrations. These straight line integrations were typically solved via numerical or symbolic integration, often leading to computationally expensive implementations. Here, a recursive method is employed to iteratively solve the integrals starting from lower order terms which are more simply evaluated. In this way, the integrals are able to be presented in a full closed-form solution. The semi-analytical boundary method is thus made fully analytical. These calculations were verified in their accuracy on a simple test case, and further verified for their generality and robustness via an intentionally convoluted test case.

Having solved the issue of implementing the semi-analytical boundary method from a numerical point of view, the focus was then turned to its implementation from a physical, modelling point of view. The semi-analytical method with no adjustments actually falls quite short of the desired accuracy for many standard test cases, such as a hydrostatic box of fluid and 3D dam break against a fixed obstacle. Particularly for the case of a hydrostatic box of fluid, two key inaccuracies were surmised. One of these was the use of an approximation to $\nabla\gamma$ used in the standard semi-analytical implementation. Though the method developed in Chapter 4 was fully accurate for both γ and $\nabla\gamma$, the standard semi-analytical implementation only uses an approximation for $\nabla\gamma$, and in a hydrostatic box of fluid it was found that this approximation was inaccurate to around 20%–50% near boundary edges, and over 100% near boundary corners. This may lead to the generation of non-physical forces and thus velocities near the edges of the hydrostatic box, developing artificial secondary currents in what should other be a completely static fluid. The second issue, though only partly addressed in this work, is the use of the standard time-discretised continuity equation. Indeed, in a semi-analytical framework, this formulation of the continuity equation tends to underestimate the density near a boundary, reducing containment and leading to fluid particles leaking from the domain. This issue appears only to be a problem when particles remain in contact with a boundary for a long time and when there is no other dominant motion in the system. This is because the same issue is not observed when simulating a transient problem such as a dam break. Since these transient problems were the original focus of this thesis, it was not deemed necessary to implement a time-independent continuity equation. Though, it is noted that when using the semi-analytical boundary method, the standard time discretisation of the continuity equation should be considered unfeasible for any long time scale simulations involving low velocities.

For the 3D transient problems of interest, such as a dam break against a fixed obstacle, the standard semi-analytical boundary implementation still suffered inaccuracies and large amounts of leaking particles. This made simulations of interest wholly unfeasible. By applying a simple particle shifting technique appropriately updated for the semi-analytical framework, as well as a new pressure assignment at the wall accounting for the normal fluid flux, accuracy of simulations and the containment of the method was vastly improved. Further, by appropriately setting the distance of the fluid particles from the boundary, pressure waves at the start of the simulation were avoided and the fluid remained stable throughout. These adjustments are straightforward to implement and render the semi-analytical boundary method much more feasible for the high speed transient problems of interest. Preliminary simulations were made of the standard 3D dam break test case were made, and the results were found to have satisfactory agreement with the experimental data. When comparing with updated boundary methods in the literature, the current implementation performed almost as well as the modified dynamic boundary condition [49] utilised in DualSPHysics, and even better than one of the standard 3D semi-analytical boundary methods in the literature [105]. These results are very promising for the value of this implementation in the wider SPH community, but it is noted that the current convergence results were not satisfactory, showing a loss in accuracy at higher resolutions, and motivating the need to quantify and understand this behaviour more clearly.

In the final chapter, the modelling of mobile boundaries, particularly those of a similar length scale to the fluid was addressed. The approaches to modelling mobile boundaries within SPH and the different regimes that must be considered were discussed, as well as coupling with the Discrete Element Method (DEM) for solid-solid interactions. The DEM method itself is well researched and validated in the literature, and as such this was left as a brief discussion only. The mesoscale modelling of fluid-solid interactions was identified as a particularly tricky regime: not large enough to fully discretise the solid bodies, but not small enough to approximate the solid as a non-Newtonian fluid. The governing equations of a two-phase mesoscale model were presented, and further to typical implementations, a formulation of the lift force within an SPH context was presented. Despite difficulties in attaining consistent validations for the lift force and other mesoscale fluid-solid interactions, it was observed that the inclusion of an inertial lift force did increase saltation rates in a test case involving 3D periodic channel flow over a bed of small DEM particles.

The main achievement in this chapter was the use of the semi-analytical formulations for spherical boundaries in Chapter 4 to provide an improved (rather, fully accurate) calculation of the porosity. This work was also published during the course of this thesis. The porosity

- the local ratio of fluid volume to total volume - is a key component of the two-phase mesoscale model. Until this point, its calculation has always been carried out via a discrete approximation through a pointwise sum over the volume of local mobile bodies. This works well when solid bodies are small enough, but quickly falls apart as the solid bodies increase in size toward the fluid resolution. It is for this reason, that typical mesoscale models in SPH require the DEM particles to be at most half the size of the fluid resolution. Beyond this point, simulations quickly become unstable and unfeasible. Noticing that the calculation of γ for spherical boundaries is equivalent to an integral over spherical volumes, it was possible to equate γ for spherical boundaries to the porosity due to local spherical bodies. As such, the porosity can be calculated fully analytically and exactly at any resolution. This does not require a full semi-analytical boundary implementation, only its application to spherical boundaries, which are much simpler and fully closed-form with simple formulae. Using exact calculations, simulations can remain stable even with DEM particles at the 90% the size of the fluid particles - something previously impossible.

7.1 Future work

Much of the work presented in this thesis is essentially introductory and many sections on their own provide opportunities for further research and development.

A potential advantage of the semi-analytical boundary method which was briefly discussed, is the opportunity to avoid discretisation of the wall boundaries entirely. The method presented in Chapter 4 does not require any wall discretisation - only wall vertices and normals. Avoiding wall particles entirely could allow for much more complex boundary shapes without concern over the quality of discretisation and placement of wall particles - something that often hinders ghost particle methods. Unfortunately, the only existing semi-analytical framework which disregards wall particles entirely does not produce satisfactory results. In fact, the method's accuracy greatly improves by adding a wall particle discretisation that allows more accurate calculation of the interactions at the wall. It would be desirable if a full and accurate framework for the semi-analytical boundary method which does not rely on wall particles could be developed. Combining this with the presented work in Chapters 4 and 5 may allow the potential for a very robust and computationally cheap boundary method. However, any attempt to remove the dependency on wall particles would require an alternative approach to approximate wall variable values over the wall, and this is likely where any attempts to progress in this area should be focused. The alternative semi-analytical boundary method which extends the fluid variables into the wall domain and integrates over them there was not fully assessed, but may provide some opportunities to leverage the work

presented in this thesis. Particularly at the point of integrating over the kernel function or its derivative - a problem that has been effectively solved here. This is largely conjecture, and more research would need to be done, likely focused on the semi-analytical governing equations when defined at the wall.

The full solutions to the normalisation terms were presented for planar and spherical boundaries. There is also the opportunity to extend the work to elliptical, cylindrical, and other shaped boundaries to further extend the model's utility. It is often the case, especially in coastal dynamics which can involve irregular grain shapes and sizes, that utilising more realistic, non-spherical solid shapes is desired.

Many of the simulations and test cases presented in this thesis are rudimentary and the analysis is only carried out at a cursory level to present initial ideas.

Further simulations could be carried out both to benchmark and assess the accuracy improvements to the semi-analytical boundary method, especially in comparison to semi-analytical implementations in literature which utilise computationally expensive implicit matrix inversions for increased accuracy. The implementation of a time-independent continuity equation could further increase the accuracy of the suggested implementation at relatively low cost. Recent improvements to the particle shifting method that do not rely on implicit matrix inversions to calculate the free surface have been developed [79] and could be extended to a semi-analytical framework. Combining these further improvements may allow for a semi-analytical method which attains similar accuracies as the state-of-the-art in the literature whilst performing at much better speeds due to avoiding computationally expensive implicit calculations. The implementation should also be validated against other standard test cases which involve highly accurate boundary modelling, such as 3D wave sloshing. Further work still has to be done on the convergence of the method. This could be done by analysing validation simulations with clearly defined analytical solutions. One such example is 3D periodic Poiseuille flow. These results are typically well-developed in their use of assessing accuracy and convergence of numerical methods and are an important step of validating any new SPH implementation.

Correcting the error associated with the approximation of $\nabla\gamma_{iw}$ as $W_{iw}\mathbf{n}_w dS$ is feasible but does require some dedicated work. In order to replace the approximation with an exact analytical solution, $\nabla\gamma$ needs to be calculated not based on the original wall boundaries and their vertices, but rather on the discretised wall boundaries due to the presence of wall particles. Each wall particle discretises a small area of the wall boundary it is attached to, which can be delineated through the use of virtual vertices belonging to each wall particle. Whilst calculating the virtual discretisation vertices for each wall particle is technically fea-

sible, it does require additional calculations and careful geometry considerations at the time of simulation set-up. Once these are deduced, it is trivial to apply the method discussed in Chapter 5, now recalculating $\nabla\gamma_{iw}$ over each individual wall particle's vertices and summing the contributions appropriately. It is not clear how computationally expensive this will be. Currently the semi-analytical boundary method is not observed to add much computational overhead in comparison to other boundary methods, but if calculations have to be added for every single interaction between a fluid particle and wall particle then this may become problematic. At this stage, the relative trade-off between improved accuracy and computational expense would have to be weighed.

The mobile boundary formulation involving drag forces, lift forces, and a fully accurate porosity could be validated and applied to many more interesting test cases. It is often difficult to find clear and consistent validation data for these types of mesoscale fluid-solid coupled flows, but nonetheless there are many problems of interest to which the framework could be applied. Relevant to the initial motivation for this thesis, applying the model to test cases involving a dam break over solid mobile bodies at a range of sizes would be particularly relevant and starting the path towards leading to improved insights on the interactions between fluids and solids during coastal wave events. The ability to model solids at a range of length scales could be leveraged to see how varying particle diameters and their combinations affect flow characteristics in typical dam break or wave flows. There is also scope to extend the model by further including not only the lift force, but also the Basset force and added-mass force, which have otherwise not been addressed in the literature for SPH-DEM. Simple validations or test cases could be used to ascertain in which regimes these additional forces are necessary, and whether they improve the quality of certain simulations as has been suggested, or they can indeed be disregarded for most applications. Review of the literature revealed a lack of high quality, reproducible experimental data for the purpose of validating mesoscale fluid-solid interactions. Simple lab scale experiments involving water flow over solids such as pebbles, gravel, and small boulders would prove invaluable in furthering the quality of numerical modelling in this domain.

A natural extension of the work presented is to apply the semi-analytical model not only to small DEM particles for porosity calculations, but also for standard large DEM particles which require their hydrodynamics forcings to be calculated based on the solid-fluid interaction terms in the governing equations. This is typically always done via embedded wall particles, and so there is a great opportunity to assess whether the semi-analytical framework can perform at acceptable levels without the use of full body discretisation (i.e.

only surface wall particles). An open question that was briefly mentioned is how the forcings on the wall particles should be reflected. If this is addressed, then this may provide the first opportunity for using the semi-analytical boundary method to model large solid bodies within a fluid flow.

8 References

- [1] Andrew D. Hwang (<https://math.stackexchange.com/users/86418/andrew-d-hwang>). *Area between geodesic and arc of constant latitude*. Mathematics Stack Exchange. <https://math.stackexchange.com/q/2115873> (version: 2017-01-28).
- [2] R.A. Dalrymple A.J.C. Crespo M. Gómez-Gesteira. “Boundary conditions generated by dynamic particles in SPH methods”. In: *Computers, Materials and Continua* 5.3 (2007), pp. 173–184.
- [3] S. Adami, X.Y. Hu, and N.A. Adams. “A generalized wall boundary condition for smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075.
- [4] Zaid Alhusban and Manousos Valyrakis. “Assessing and Modelling the Interactions of Instrumented Particles with Bed Surface at Low Transport Conditions”. In: *Applied Sciences* 11.16 (2021).
- [5] Sk Zeeshan Ali and Subhasish Dey. “Bed particle saltation in turbulent wall-shear flow: A review”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 475.2223 (2019).
- [6] Andrea Amicarelli et al. “A Smoothed Particle Hydrodynamics model for 3D solid body transport in free surface flows”. In: *Computers and Fluids* 116 (2015), pp. 205–228.
- [7] Andrea Amicarelli et al. “SPH truncation error in estimating a 3D function”. In: *Computers and Fluids* 44.1 (2011), pp. 279–296.
- [8] T. B. Anderson and Roy Jackson. “Fluid Mechanical Description of Fluidized Beds. Equations of Motion”. In: *Industrial & Engineering Chemistry Fundamentals* 6 (4 Nov. 1967), pp. 527–539.
- [9] M. Antuono, A. Colagrossi, and S. Marrone. “Numerical diffusive terms in weakly-compressible SPH schemes”. In: *Computer Physics Communications* 183.12 (2012), pp. 2570–2580.
- [10] M. Antuono et al. “A measure of spatial disorder in particle methods”. In: *Computer Physics Communications* 185.10 (2014), pp. 2609–2621.
- [11] M. Antuono 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.
- [12] D. A. Barbosa and F. P. Piccoli. “Comparing the force due to the Lennard-Jones potential and the Coulomb force in the SPH Method”. In: *Journal of Ocean Engineering and Science* 3.4 (2018), pp. 310–315.
- [13] Daniel James Barker. “The application of smoothed particle hydrodynamics to the simulation of multiphase flows through packed beds”. In: December (2015).

-
- [14] A. Barreiro et al. “Smoothed Particle Hydrodynamics for coastal engineering problems”. In: *Computers & Structures* 120 (2013), pp. 96–106.
 - [15] G. Bartzke et al. “Simulation of flow past a sphere on a rough bed using smoothed particle hydrodynamics (SPH)”. In: *Computational Particle Mechanics* 9 (Sept. 2022), pp. 927–940.
 - [16] Laurence Bergougnoux et al. “The motion of solid spherical particles falling in a cellular flow field at low Stokes number”. In: *Physics of Fluids* 26 (9 Sept. 2014).
 - [17] J. Bonet and T.-S.L. Lok. “Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations”. In: *Computer Methods in Applied Mechanics and Engineering* 180.1 (1999), pp. 97–115.
 - [18] Boost. *Boost C++ Libraries*. <http://www.boost.org/>. Last accessed version 1.67, April 2018. 2022.
 - [19] Boost. *Gauss-Legendre quadrature*. https://www.boost.org/doc/libs/1_67_0/libs/math/doc/html/math_toolkit/gauss.html. Last accessed version 1.67, April 2018. 2018.
 - [20] Mariamawit Borga et al. “A geospatial framework to estimate depth of scour under buildings due to storm surge in coastal areas”. In: *Natural Hazards* 87.3 (July 2017), pp. 1285–1311.
 - [21] Simon Burchett et al. *A Standard Approach to Coastal Structures?* Aug. 2018, pp. 513–528.
 - [22] J. Calderon-Sanchez, J.L. Cercos-Pita, and D. Duque. “A geometric formulation of the Shepard renormalization factor”. In: *Computers & Fluids* 183 (Apr. 2019), pp. 16–27.
 - [23] R.B. Canelas et al. “Extending DualSPHysics with a Differential Variational Inequality: modeling fluid-mechanism interaction”. In: *Applied Ocean Research* 76 (July 2018), pp. 88–97.
 - [24] Ricardo B. Canelas et al. “DualSPHysics Current Developments: SPH-DCDEM”. In: *Dual-SPHysics User Workshop*. Sept. 2015.
 - [25] Ricardo B. Canelas et al. “SPH-DCDEM model for arbitrary geometries in free surface solid–fluid flows”. In: *Computer Physics Communications* 202 (May 2016), pp. 131–140.
 - [26] J.L. Cercos-Pita. “AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL”. In: *Computer Physics Communications* 192 (2015), pp. 295–312.
 - [27] Jie Chen et al. “Laboratory study on protection of tsunami-induced scour by offshore breakwaters”. In: *Natural Hazards* 81.2 (Mar. 2016), pp. 1229–1247.
 - [28] L. Chiron et al. “Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows”. In: *Computer Physics Communications* 234 (Jan. 2019), pp. 93–111.
 - [29] Coalition for Urban Transitions. *Climate Emergency, Urban Opportunity*. Tech. rep. Coalition for Urban Transitions, 2019, p. 160.

- [30] Andrea Colagrossi and Maurizio Landrini. “Numerical simulation of interfacial flows by smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 191.2 (Nov. 2003), pp. 448–475.
- [31] Andrea Colagrossi et al. “Particle packing algorithm for SPH schemes”. In: *Computer Physics Communications* 183 (8 Aug. 2012), pp. 1641–1653.
- [32] Peter J. Cossins. “Smoothed Particle Hydrodynamics”. In: *Instrumentation and Methods for Astrophysics* (July 2010).
- [33] A.J.C. Crespo et al. “DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)”. In: *Computer Physics Communications* 187 (Feb. 2015), pp. 204–216.
- [34] Alejandro C. Crespo et al. “GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods”. In: *PLoS ONE* 6.6 (June 2011). Ed. by Jörg Langowski, e20685.
- [35] S. J. Cummins, T. B. Silvester, and P. W. Cleary. “Three-dimensional wave impact on a rigid structure using smoothed particle hydrodynamics”. In: *International Journal for Numerical Methods in Fluids* 68.12 (Apr. 2012), pp. 1471–1496.
- [36] R.A. Dalrymple and Omar Knio. “SPH modelling of water waves”. In: *Coastal Dynamics 2001*. Reston, VA: American Society of Civil Engineers, May 2001, pp. 779–787.
- [37] R.A. Dalrymple and B.D. Rogers. “Numerical modeling of water waves with the SPH method”. In: *Coastal Engineering* 53.2 (2006). Coastal Hydrodynamics and Morphodynamics, pp. 141–147.
- [38] M. De Leffe, D. Le Touzé, and B. Alessandrini. “Normal flux method at the boundary for SPH”. In: *4th Int. SPHERIC Workshop (SPHERIC 2009)*. Nantes, France, 2009.
- [39] C. Derschum et al. “Debris impact under extreme hydrodynamic conditions part 1: Hydrodynamics and impact geometry”. In: *Coastal Engineering* 141.August (2018), pp. 24–35.
- [40] Antonio Di Monaco et al. “SPH modeling of solid boundaries through a semi-analytic approach”. In: *Engineering Applications of Computational Fluid Mechanics* 5.1 (2011), pp. 1–15.
- [41] Gary A. Dilts. “Moving-Least-Squares-particle hydrodynamics - I. Consistency and stability”. In: *International Journal for Numerical Methods in Engineering* 44.8 (Mar. 1999), pp. 1115–1155.
- [42] J.M. Domínguez et al. “DualSPHysics: from fluid dynamics to multiphysics problems”. In: *Computational Particle Mechanics* 0123456789 (2021).
- [43] J.M. Domínguez et al. “Neighbour lists in smoothed particle hydrodynamics”. In: *International Journal for Numerical Methods in Fluids* 67.12 (Dec. 2011), pp. 2026–2042.

-
- [44] J.M. Domínguez et al. “Towards a more complete tool for coastal engineering: Solitary wave generation, propagation and breaking in an SPH-based model”. In: *Coastal Engineering Journal* 61.1 (2019), pp. 15–40.
 - [45] Jose M. Domínguez, Alejandro J.C. Crespo, and Moncho Gómez-Gesteira. “Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method”. In: *Computer Physics Communications* 184.3 (2013), pp. 617–627.
 - [46] EEFIT. *Recovery Two Years after the 2011 Tōhoku Earthquake and Tsunami: A Return Mission Report by EEFIT*. Tech. rep. EEFIT, 2013, p. 160.
 - [47] EERI Special Earthquake Report. *The Japan Tohoku Tsunami of March 11, 2011*. Tech. rep. EERI, 2011.
 - [48] Office of Electricity Delivery and U.S. Department of Energy Energy Reliability. *Effect of Sea Level Rise on Energy Infrastructure in Four Major Metropolitan Areas*. Tech. rep. September. 2014, p. 44.
 - [49] A. English et al. “Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): application to tank sloshing, dam break and fish pass problems”. In: *Computational Particle Mechanics* (2021).
 - [50] “Experimental observations of tsunami induced scour at onshore structures”. In: *Coastal Engineering* 152 (Oct. 2019), p. 103505.
 - [51] R. Fatehi and M. T. Manzari. “Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives”. In: *Computers and Mathematics with Applications* 61.2 (Jan. 2011), pp. 482–498.
 - [52] J Feldman and J Bonet. “Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems”. In: March 2007 (2010), pp. 295–324.
 - [53] M. Ferrand 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 (Feb. 2013), pp. 446–472.
 - [54] Martin Ferrand et al. “Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D”. In: *Computer Physics Communications* 210 (2017), pp. 29–44.
 - [55] *Flood Resistant Design and Construction*. ASCE/SEI 24-14. American Society of Civil Engineers, 2014.
 - [56] G Fourtakas, B D Rogers, and D Laurence. “3-D SPH Modelling of Sediment Scouring Induced by Rapid Flows”. In: *Proceedings of the 9th SPHERIC International Workshop* June (2014), pp. 9–16.

- [57] G Fourtakas and B. D. BD Rogers. “Modelling multi-phase liquid-sediment scour and re-suspension induced by rapid flows using Smoothed Particle Hydrodynamics (SPH) accelerated with a Graphics Processing Unit (GPU)”. In: *Advances in Water Resources* 92 (2016), pp. 186–199.
- [58] Georgios Fourtakas et al. “Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models”. In: *Computers and Fluids* 190 (2019), pp. 346–361.
- [59] Mats van Gemert. “Evaluation of new Eurocode ‘prEN1991-1-8’ on wave and current actions: Case study IJmuiden - Does the new Eurocode improve the current design practice of breakwaters and other coastal structures?” MA thesis. Delft University of Technology, Mar. 2022.
- [60] Ahmed Ghobarah, Murat Saatcioglu, and Ioan Nistor. “The impact of the 26 December 2004 earthquake and tsunami on structures and infrastructure”. In: *Engineering Structures* 28.2 (Jan. 2006), pp. 312–326.
- [61] Dimitri Gidaspow, Rukmini Bezburaah, and Jianmin Ding. “Hydrodynamics of circulating fluidized beds: Kinetic theory approach”. In: 1992.
- [62] R.A. Gingold and J. J. Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389.
- [63] M. Gomez-Gesteira et al. “SPHysics – development of a free-surface fluid solver – Part 1: Theory and formulations”. In: *Computers & Geosciences* 48 (2012), pp. 289–299.
- [64] Hitoshi Gotoh, Songdong Shao, and Tetsu Memita. “SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater”. In: *Coastal Engineering Journal* 46.1 (2004), pp. 39–63.
- [65] Giang D. Nguyen Ha H. Bui. “A coupled fluid-solid SPH approach to modelling flow through deformable porous media”. In: *International Journal of Solids and Structures* 125 (Oct. 2017), pp. 244–264.
- [66] Takahiro Harada, Seichi Koshizuka, and Yoichiro Kawaguchi. “Smoothed Particle Hydrodynamics on GPUs”. In: *Computer Graphics International* (2007), pp. 63–70.
- [67] Adam Hatzikyriakou et al. “Component-Based Vulnerability Analysis for Residential Structures Subjected to Storm Surge Impact from Hurricane Sandy”. In: *Natural Hazards Review* 17.1 (Feb. 2016), p. 05015005.
- [68] Yi He et al. “A GPU-based coupled SPH-DEM method for particle-fluid flow with free surfaces”. In: *Powder Technology* 338 (Oct. 2018), pp. 548–562.

-
- [69] Alexis H  rault, Giuseppe Bilotta, and Robert A. Dalrymple. “SPH on GPU with CUDA”. In: *Journal of Hydraulic Research* 48.SUPPL. 1 (2010), pp. 74–79.
 - [70] Gene Hou, Jin Wang, and Anita Layton. “Numerical Methods for Fluid-Structure Interaction — A Review”. In: *Communications in Computational Physics* 12.2 (2012), pp. 337–377.
 - [71] Trevor Houser and Peter Marsters. *The World’s Second Largest Blackout*. Tech. rep. Rhodium Group, 2018.
 - [72] Xiang Yu Hu and N. A. Adams. “A multi-phase SPH method for macroscopic and mesoscopic flows”. In: *Journal of Computational Physics* 213.2 (Apr. 2006), pp. 844–861.
 - [73] Jason P. Hughes and David I. Graham. “Comparison of incompressible and weakly-compressible SPH models for free-surface water flows”. In: *Journal of Hydraulic Research* 48.sup1 (Jan. 2010), pp. 105–117.
 - [74] Fumihiko Imamura, Kazuhisa Goto, and Shigeki Ohkubo. “A numerical model for the transport of a boulder by tsunami”. In: *Journal of Geophysical Research: Oceans* 113.C1 (2008).
 - [75] British Standards Institution. *Maritime structures – Part 7: Guide to the design and construction of breakwaters (incorporating corrigendum January 2010)*. London, UK: BSI Standards Limited, 1991. 94 pp.
 - [76] IPCC. Technical Summary. In: *IPCC Special Report on the Ocean and Cryosphere in a Changing Climate*. Ed. by H.-O. P  rtner et al. In press. 2019.
 - [77] Richard M. Iverson. “Scaling and design of landslide and debris-flow experiments”. In: *Geomorphology* 244 (Sept. 2015), pp. 9–20.
 - [78] Richard M. Iverson et al. “The perfect debris flow? Aggregated results from 28 large-scale experiments”. In: *Journal of Geophysical Research* 115.F3 (July 2010).
 - [79] Mojtaba Jandaghian, Herman Musumari Siaben, and Ahmad Shakibaeinia. “Stability and accuracy of the weakly compressible SPH with particle regularization techniques”. In: *European Journal of Mechanics, B/Fluids* 94 (July 2022), pp. 314–333.
 - [80] Ravindra Jayaratne et al. “Predictive model for scour depth of coastal structure failures due to tsunamis”. In: *Proceedings of the Coastal Engineering Conference*. Vol. 2014-Janua. 2014.
 - [81] K.M.T. Kleefsman et al. “A Volume-of-Fluid based simulation method for wave impact problems”. In: *Journal of Computational Physics* 206.1 (2005), pp. 363–393.
 - [82] H.T.-S. Ko et al. “Hydraulic Experiments on Impact Forces from Tsunami-Driven Debris”. In: *Journal of Waterway, Port, Coastal, and Ocean Engineering* 141.3 (2015), p. 04014043.
 - [83] Wawrzyniec Kostorz and Anton Esmail-Yakas. “A semi-analytical boundary integral method for radial functions with application to Smoothed Particle Hydrodynamics”. In: *Journal of Computational Physics* 417 (Sept. 2020).

- [84] Wawrzyniec Kostorz and Anton Esmail-Yakas. “Semi-analytical smoothed-particle hydrodynamics correction factors for polynomial kernels and piecewise-planar boundaries”. In: *International Journal for Numerical Methods in Engineering* 122 (24 Dec. 2021), pp. 7271–7305.
- [85] Wawrzyniec Kostorz and Anton Esmail-Yakas. “Semi-analytical treatment of spherical solids in smoothed-particle hydrodynamics fluid simulations”. In: *International Journal for Numerical Methods in Engineering* 122 (7 Apr. 2021), pp. 1732–1745.
- [86] S. Kulasegaram et al. “A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications”. In: *Computational Mechanics* 33.4 (Mar. 2004), pp. 316–325.
- [87] Bjarke Eltard Larsen et al. “Experimental study of tsunami-induced scour around a monopile foundation”. In: *Coastal Engineering* 138 (Aug. 2018), pp. 9–21.
- [88] Bjarke Eltard Larsen et al. “Tsunami-induced scour around monopile foundations”. In: *Coastal Engineering* 129 (Nov. 2017), pp. 36–49.
- [89] Linlin Li et al. “A modest 0.5-m rise in sea level will double the tsunami hazard in Macau”. In: *Science Advances* 4.8 (2018), eaat1180.
- [90] S.J. Lind et al. “Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves”. In: *Journal of Computational Physics* 231.4 (2012), pp. 1499–1523.
- [91] Steven J. Lind, Benedict D. Rogers, and Peter K. Stansby. “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).
- [92] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics*. WORLD SCIENTIFIC, Oct. 2003.
- [93] M.B. Liu, G.R. Liu, and K.Y. Lam. “Constructing smoothing functions in smoothed particle hydrodynamics with applications”. In: *Journal of Computational and Applied Mathematics* 155.2 (2003), pp. 263–284.
- [94] L. Lobovský et al. “Experimental investigation of dynamic pressure loads during dam break”. In: *Journal of Fluids and Structures* 48 (June 2014), pp. 407–434.
- [95] L.B. Lucy. “A numerical approach to the testing of the fission hypothesis”. In: *The Astrophysical Journal* 82 (Dec. 1977), pp. 1013–1024.
- [96] Hongwang Ma, Jun Yang, and Longzhu Chen. “Effect of scour on the structural response of an offshore wind turbine supported on tripod foundation”. In: *Applied Ocean Research* 73 (Apr. 2018), pp. 179–189.

-
- [97] M. Mabssout and M. I. Herreros. “Runge-Kutta vs Taylor-SPH: Two time integration schemes for SPH with application to Soil Dynamics”. In: *Applied Mathematical Modelling* 37.5 (2013), pp. 3541–3563.
 - [98] Fabricio Maciá et al. “Theoretical analysis of the no-slip boundary condition enforcement in SPH methods”. In: *Progress of Theoretical Physics* 125.6 (June 2011), pp. 1091–1121.
 - [99] S. Marrone et al. “ δ -SPH model for simulating violent impact flows”. In: *Computer Methods in Applied Mechanics and Engineering* 200.13-16 (Mar. 2011), pp. 1526–1542.
 - [100] S. Marrone et al. “Study of ship wave breaking patterns using 3D parallel SPH simulations”. In: *Computers and Fluids* 69 (2012), pp. 54–66.
 - [101] Jeff Masters. *How sea level rise contributes to billions in extra damage during hurricanes*. 2022. URL: <https://yaleclimateconnections.org/2022/10/how-sea-level-rise-contributes-to-billions-in-extra-damage-during-hurricanes/> (visited on 04/02/2023).
 - [102] Jeff Masters and Bob Henson. *The U.S. had 18 different billion-dollar weather disasters in 2022*. 2023. URL: <https://yaleclimateconnections.org/2023/01/the-u-s-had-18-different-billion-dollar-weather-disasters-in-2022/> (visited on 04/02/2023).
 - [103] Clara Matutano et al. “Scour prediction and scour protections in offshore wind farms”. In: *Renewable Energy* 57 (Sept. 2013), pp. 358–365.
 - [104] Arno Mayrhofer et al. “Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions”. In: *Computer Physics Communications* 184.11 (2013), pp. 2515–2527.
 - [105] Arno Mayrhofer et al. “Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D”. In: *Numerical Algorithms* 68.1 (Jan. 2015), pp. 15–34.
 - [106] Farrukh Mazhar et al. “On the meshfree particle methods for fluid-structure interaction problems”. In: *Engineering Analysis with Boundary Elements* 124 (2021), pp. 14–40.
 - [107] Julie McNamara et al. *Lights Out? Storm Surge, Blackouts and How Clean Energy Can Help*. Tech. rep. October. 2015, pp. 1–44.
 - [108] Domenica Mirauda et al. “Smoothed particle hydrodynamics modeling with advanced boundary conditions for two-dimensional dam-break floods”. In: *Water (Switzerland)* 12 (4 Apr. 2020).
 - [109] Hans de Moel, Wouter Botzen, and Jeroen Aerts. “Economic and direct losses from Hurricane Sandy”. In: *Annals of the New York Academy of Sciences* 1294 (2013), pp. 81–89.
 - [110] Athanasios Mokos, Benedict D. Rogers, and Peter K. Stansby. “A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles”. In: *Journal of Hydraulic Research* 55 (2 Mar. 2017), pp. 143–162.

- [111] Diego Molteni and Andrea Colagrossi. “A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH”. In: *Computer Physics Communications* 180 (2009), pp. 861–872.
- [112] J. J. Monaghan. “Simulating Free Surface Flows with SPH”. In: *Journal of Computational Physics* 110.2 (1994), pp. 399–406.
- [113] J. J. Monaghan. “Smoothed Particle Hydrodynamics”. In: *Annual Review of Astronomy and Astrophysics* 30.1 (1992), pp. 543–574.
- [114] J. J. Monaghan. “Smoothed particle hydrodynamics”. In: *Reports on Progress in Physics* 68.8 (2005), pp. 1703–1759.
- [115] J. J. Monaghan. “Smoothed Particle Hydrodynamics and Its Diverse Applications”. In: *Annual Review of Fluid Mechanics* 44.1 (2012), pp. 323–346.
- [116] J. J. Monaghan. “SPH compressible turbulence”. In: *Monthly Notices of the Royal Astronomical Society* 335.3 (2002), pp. 843–852.
- [117] J. J. Monaghan and J. B. Kajtár. “SPH particle boundary forces for arbitrary boundaries”. In: *Computer Physics Communications* 180.10 (2009), pp. 1811–1820.
- [118] J. J. Monaghan and J. C. Lattanzio. “A refined particle method for astrophysical problems”. In: *Astronomy and astrophysics* 149.1 (1985), pp. 135–143.
- [119] Joseph P. Morris, Patrick J. Fox, and Yi Zhu. “Modeling Low Reynolds Number Incompressible Flows Using SPH”. In: *Journal of Computational Physics* 136.1 (1997), pp. 214–226.
- [120] Clay Naito et al. “Procedure for Site Assessment of the Potential for Tsunami Debris Impact”. In: *Journal of Waterway, Port, Coastal, and Ocean Engineering* 140.2 (2014), pp. 223–232.
- [121] Helio A. Navarro and Meire P. de Souza Braun. “Determination of the normal spring stiffness coefficient in the linear spring-dashpot contact model of discrete element method”. In: *Powder Technology* 246 (2013), pp. 707–722.
- [122] Stephen J Neethling and Daniel J Barker. “Using Smooth Particle Hydrodynamics (SPH) to model multiphase mineral processing systems”. In: *Minerals Engineering* 90 (2016), pp. 17–28.
- [123] Robert J. Nicholls, Frank M.J. Hoozemans, and Marcel Marchand. “Increasing flood risk and wetland losses due to global sea-level rise: regional and global analyses”. In: *Global Environmental Change* 9 (Oct. 1999), S69–S87.
- [124] Ioan Nistor, Nils Goseberg, and Jacob Stolle. “Tsunami-Driven Debris Motion and Loads: A Critical Review”. In: *Frontiers in Built Environment* 3 (Jan. 2017), p. 2.
- [125] Younes Nouri et al. “Experimental investigation of tsunami impact on free standing structures”. In: *Coastal Engineering Journal* 52.1 (2010), pp. 43–70.

-
- [126] G. Oger et al. “Hybrid CPU-GPU acceleration of the 3-D parallel code SPH-Flow”. In: *5th International SPHERIC Workshop*. Manchester, UK, June 2010.
 - [127] G. Oger et al. “On distributed memory MPI-based parallelization of SPH codes in massive HPC context”. In: *Computer Physics Communications* 200 (Mar. 2016), pp. 1–14.
 - [128] Christina M. Patricola and Michael F. Wehner. “Anthropogenic influences on major tropical cyclone events”. In: *Nature* 563.7731 (Nov. 2018), pp. 339–346.
 - [129] Chong Peng et al. “Multiphase SPH modeling of free surface flow in porous media with variable porosity”. In: *Computers and Geotechnics* 81 (2017), pp. 239–248.
 - [130] Daniel J. Price. “Smoothed particle hydrodynamics and magnetohydrodynamics”. In: *Journal of Computational Physics* 231 (3 Feb. 2012), pp. 759–794.
 - [131] Daniel J. Price. *Smoothed Particle Hydrodynamics: Things I wish my mother taught me*. Nov. 2011.
 - [132] Nathan J. Quinlan, M. Basa, and M. Lastiwka. “Truncation error in mesh-free particle methods”. In: *International Journal for Numerical Methods in Engineering* 66 (13 Jan. 2006), pp. 2064–2085.
 - [133] Alison Raby et al. “Implications of the 2011 Great East Japan Tsunami on sea defence design”. In: *International Journal of Disaster Risk Reduction* 14.4 (2015), pp. 332–346.
 - [134] H. R. Riggs et al. “Experimental and Analytical Study of Water-Driven Debris Impact Forces on Structures”. In: *Journal of Offshore Mechanics and Arctic Engineering* 136.4 (2014), pp. 1–8.
 - [135] A. Ritter. “Die Fortpflanzung der Wasserwellen”. In: *Zeitschrift des Vereines Deutscher Ingenieure* 36 (1892), pp. 947–954.
 - [136] Martin Robinson, Marco Ramaoli, and Stefan Luding. “Fluid-particle flow simulations using two-way-coupled mesoscale SPH-DEM and validation”. In: *International Journal of Multiphase Flow* 59 (Feb. 2014), pp. 121–134.
 - [137] P. G. Saffman. “The lift on a small sphere in a slow shear flow”. In: *Journal of Fluid Mechanics* 22.2 (1965), pp. 385–400.
 - [138] Christian Scheidl et al. “Analysing Debris-Flow Impact Models, Based on a Small Scale Modelling Approach”. In: *Surveys in Geophysics* 34.1 (2013), pp. 121–140.
 - [139] Hans F. Schwaiger. “An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions”. In: *International Journal for Numerical Methods in Engineering* 75.6 (Aug. 2008), pp. 647–671.
 - [140] Imperial College London Research Computing Service. *Research Computing Service - High Performance Computing*. Online. Accessed: 2022-06-11.

- [141] M.S. Shadloo, G. Oger, and D. Le Touzé. “Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, Current state, And challenges”. In: *Computers and Fluids* 136 (Sept. 2016), pp. 11–34.
- [142] Seyedreza Shafiei et al. “Measurements of tsunami-borne debris impact on structures using an embedded accelerometer”. In: *Journal of Hydraulic Research* 54.4 (2016), pp. 435–449.
- [143] S. Shahriari, I.G. Hassan, and L. Kadem. “Modeling unsteady flow characteristics using smoothed particle hydrodynamics”. In: *Applied Mathematical Modelling* 37.3 (2013), pp. 1431–1450.
- [144] Krishnendu Shekhar et al. “Conceptual Evaluation of Tsunami Debris Field Damming and Impact Forces”. In: *Journal of Waterway, Port, Coastal, and Ocean Engineering* 146.6 (2020), p. 04020039.
- [145] Alex Skillen et al. “Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body-water slam and efficient wave-body interaction”. In: *Computer Methods in Applied Mechanics and Engineering* 265 (2013), pp. 163–173.
- [146] Branwen E. Snelling et al. “Improvements to a smooth particle hydrodynamics simulator for investigating submarine landslide generated waves”. In: *International Journal for Numerical Methods in Fluids* 92.7 (2020), pp. 744–764.
- [147] Sangyoung Son, Patrick Lynett, and Aykut Ayca. “Modelling scour and deposition in harbours due to complex tsunami-induced currents”. In: *Earth Surface Processes and Landforms* 45.4 (2020), pp. 978–998.
- [148] S. L. Soo. “Particulates and Continuum. Multiphase Fluid Dynamics.” In: (1989).
- [149] SPHERIC - SPH rEsearch and engineeRing International Community. *Test 02. 3-D schematic dam break and evolution of the free surface*. Online. Accessed: 2022-06-11.
- [150] Daisuke Sugawara, Kazuhisa Goto, and Bruce E. Jaffe. “Numerical models of tsunami sediment transport — Current understanding and future directions”. In: *Marine Geology* 352 (2014). 50th Anniversary Special Issue, pp. 295–320.
- [151] P. N. Sun et al. “The δ plus-SPH model: Simple procedures for a further improvement of the SPH scheme”. In: *Computer Methods in Applied Mechanics and Engineering* 315.315 (Mar. 2017), pp. 25–49.
- [152] P.N. Sun et al. “Multi-resolution Delta-plus-SPH with tensile instability control: Towards high Reynolds number flows”. In: *Computer Physics Communications* 224 (Mar. 2018), pp. 63–80.

-
- [153] Ataru Tanikawa et al. “Phantom-GRAPE : Numerical software library to accelerate collisionless N -body simulation with SIMD instruction set on x86 architecture”. In: *New Astronomy* 19 (2013), pp. 74–88.
 - [154] Nikolas Sokratis Tavouktsoglou. “Scour and scour protection around offshore gravity based foundations”. PhD thesis. 2018.
 - [155] Susan Tonkin et al. “Tsunami scour around a cylinder”. In: *Journal of Fluid Mechanics* 496.496 (Dec. 2003), pp. 165–192.
 - [156] D. Le Touzé et al. “A critical investigation of smoothed particle hydrodynamics applied to problems with free-surfaces”. In: *International Journal for Numerical Methods in Fluids* 73 (7 Nov. 2013), pp. 660–691.
 - [157] Kevin E. Trenberth et al. “Hurricane Harvey Links to Ocean Heat Content and Climate Change Adaptation”. In: *Earth’s Future* 6.5 (May 2018), pp. 730–744.
 - [158] Andi Trimulyono, Hirotada Hashimoto, and Kouki Kawamura. “Experimental Validation of SPH for Wave Generation and Propagation in Large Wave Tank”. In: *Proceedings of the Twenty-seventh (2017) International Ocean and Polar Engineering Conference San Francisco, CA, USA, June 25-30*. ISOPE, July 2017, pp. 584–590.
 - [159] Renato Vacondio et al. “3D SPH numerical simulation of the wave generated by the Vajont rockslide”. In: *Italian Journal of Engineering Geology and Environment* 2013.TOPIC4 (2013), pp. 445–456.
 - [160] Renato Vacondio et al. “Grand challenges for Smoothed Particle Hydrodynamics numerical schemes”. In: *Computational Particle Mechanics* (2020).
 - [161] Damien Violeau and Benedict D. Rogers. “Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future”. In: *Journal of Hydraulic Research* 54.1 (2016), pp. 1–26.
 - [162] Zhi Bin Wang et al. “An overview of smoothed particle hydrodynamics for simulating multiphase flow”. In: *Applied Mathematical Modelling* 40.23-24 (2016), pp. 1339–1351.
 - [163] Holger Wendland. “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree”. In: *Advances in Computational Mathematics* 4.1 (Dec. 1995), pp. 389–396.
 - [164] Chester K. Wentworth. “A Scale of Grade and Class Terms for Clastic Sediments”. In: *The Journal of Geology* 30 (5 July 1922), pp. 377–392.
 - [165] Ke Wu, Dongmin Yang, and Nigel Wright. “A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure”. In: *Computers and Structures* 177 (Dec. 2016), pp. 141–161.

- [166] Yao Yao et al. “A preliminary laboratory study of motion of floating debris generated by solitary waves running up a beach”. In: *Journal of Earthquake and Tsunami* 08.03 (2014).
- [167] Ting Ye et al. “Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications”. In: *Physics of Fluids* 31 (1 Jan. 2019).
- [168] H. Yeh and H. B. Mason. “Sediment response to tsunami loading: Mechanisms and estimates”. In: *Geotechnique* 64.2 (2014), pp. 131–143.
- [169] Harry Yeh and Wenwen Li. “Tsunami scour and sedimentation”. In: *Fourth International Conference on Scour and Erosion* (2008), pp. 95–106.
- [170] Harry Yeh et al. “Mechanisms of Scour Induced by Tsunami Runup”. In: *Proceedings of the 2nd international conference on scour and erosion* (2004), pp. 464–471.
- [171] Xing Zheng, Qingwei Ma, and Songdong Shao. “Study on SPH Viscosity Term Formulations”. In: *Applied Sciences* 8.2 (Feb. 2018), p. 249.
- [172] H.P. Zhu et al. “Discrete particle simulation of particulate systems: Theoretical developments”. In: *Chemical Engineering Science* 62.13 (2007). *Frontier of Chemical Engineering - Multi-scale Bridge between Reductionism and Holism*, pp. 3378–3396.

A Full expansion of $\frac{\partial^2 A}{\partial \mathbf{r}^2} : (\mathbf{r} - \mathbf{r}') \otimes (\mathbf{r} - \mathbf{r}')$

Here we provide the full expression for the third term in Eq.(5). This term involves the double dot product and tensor product of the second-rank tensor $\frac{\partial^2 A}{\partial \mathbf{r}^2}$ and the vector $(\mathbf{r} - \mathbf{r}')$.

The double dot product is a contraction of two second-rank tensors, which produces a scalar. Specifically, for two tensors $\mathbf{A}$ and $\mathbf{B}$, the double dot product is defined as:

$$\mathbf{A} : \mathbf{B} = A_{ij}B_{ij} = \sum_{i=1}^3 \sum_{j=1}^3 A_{ij}B_{ij}.$$

The tensor product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is a second-rank tensor, which can be represented as an outer product:

$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{pmatrix}.$$

As such, the full component-by-component expansion of $\frac{\partial^2 A}{\partial \mathbf{r}^2} : (\mathbf{r} - \mathbf{r}') \otimes (\mathbf{r} - \mathbf{r}')$ can be written as

$$\frac{\partial^2 A}{\partial x_i \partial x_j} (x_i - x'_i) (x_j - x'_j) = \begin{bmatrix} \frac{\partial^2 A}{\partial x^2} (x - x')^2 & \frac{\partial^2 A}{\partial x \partial y} (x - x')(y - y') & \frac{\partial^2 A}{\partial x \partial z} (x - x')(z - z') \\ \frac{\partial^2 A}{\partial y \partial x} (y - y')(x - x') & \frac{\partial^2 A}{\partial y^2} (y - y')^2 & \frac{\partial^2 A}{\partial y \partial z} (y - y')(z - z') \\ \frac{\partial^2 A}{\partial z \partial x} (z - z')(x - x') & \frac{\partial^2 A}{\partial z \partial y} (z - z')(y - y') & \frac{\partial^2 A}{\partial z^2} (z - z')^2 \end{bmatrix} \quad (\text{A.1})$$

B Derivation of dimension reduction process for γ in 2D

In 2D, the equation for γ at a particle a can be understood as a 2D area integral over $\Omega \cap \Omega_a$, which is an area that lies in the same plane as our domain. That is

$$\gamma_a = \iint_{\Omega \cap \Omega_a} W(|\mathbf{r}_a - \mathbf{r}'|) d^2 \mathbf{r}' \quad (\text{B.1})$$

where $d^2 \mathbf{r}'$ here represents the unit area of $\Omega \cap \Omega_a$ and its vector area necessarily points “up”, or in the direction out of the 2D domain.

Stokes’ theorem can be written as

$$\iint_{\Sigma} (\nabla \times \mathbf{F}) \cdot d^2 \Sigma = \oint_{\partial \Sigma} \mathbf{F} \cdot d\Sigma \quad (\text{B.2})$$

where $\oint_{\partial\Sigma} d\Sigma$ is the 1D integral over the boundary of the area Σ . Comparing this to Eq. B.1 for γ , we can write the following equivalences

$$\begin{aligned}\Sigma &\equiv \Omega \cap \Omega_a \\ \iint_{\Sigma} &\equiv \iint_{\Omega \cap \Omega_a} \\ W(|\mathbf{r}_a - \mathbf{r}'|) d^2\mathbf{r}' &\equiv (\nabla \times \mathbf{F}) \cdot d^2\Sigma\end{aligned}$$

The key equivalency is the third one, which lets us know that if we can find a formulation for $\mathbf{F}$ which satisfies the equivalency and the constraints of our problem, then we should be able to apply Stokes' theorem to Eq. B.1 directly. For the area boundaries, is also noted that $\partial\Sigma \equiv \partial\Omega$, $\oint_{\partial\Sigma} \equiv \oint_{\partial\Omega}$, and $d\Sigma = d\mathbf{r}'$ in our notation.

We are working in a 2D domain with the radially symmetric function, W . This suggests the use of a polar coordinate system is appropriate, and we can rewrite $\rho = |\mathbf{r}_a - \mathbf{r}'|$ which is a variable in the 2D plane. As mentioned previously, we are working with vector areas which point in a single consistent “up” direction out of our domain, and so it is logical to extend our problem to three dimensions for the purpose of capturing the direction of the vector areas. A cylindrical coordinate system is then the most appropriate. This makes clear that $d^2\mathbf{r}'$ pointing out of the 2D domain is equivalent to saying it points in the z direction, $\mathbf{e}_z$. For the context of our problem, where the area Σ lies flat on the 2D domain, it also points in the $\hat{z}$ direction, and so we can write $d^2\mathbf{r}'\mathbf{e}_z \equiv d^2\Sigma$.

To satisfy the third equivalency, it is therefore sufficient that

$$\nabla \times \mathbf{F} = W(\rho)\mathbf{e}_z \quad (\text{B.3})$$

It is important to recognise from this equation that $\nabla \times \mathbf{F}$ as a function only points in the $\hat{z}$ direction and depends only on ρ , i.e. varies only in the $\mathbf{e}_\rho$ direction. This can be intuited from the symmetry of the problem, but it has now been shown explicitly here.

Considering the curl operator in cylindrical coordinates, we can assess what the form of $\mathbf{F}$ should be.

$$\nabla \times \mathbf{F} = \left(\frac{1}{\rho} \frac{\partial F_z}{\partial \phi} - \frac{\partial F_\phi}{\partial z} \right) \mathbf{e}_\rho + \left(\frac{\partial F_\rho}{\partial z} - \frac{\partial F_z}{\partial \rho} \right) \mathbf{e}_\phi + \frac{1}{\rho} \left(\frac{\partial(\rho F_\phi)}{\partial \rho} - \frac{\partial F_\rho}{\partial \phi} \right) \mathbf{e}_z \quad (\text{B.4})$$

As it only points in the $\hat{z}$ direction we can set all terms but the last equal to 0. As it should only depend on ρ , there should be no gradients with respect to any variable other than ρ as these would introduce either a z or ϕ dependency, and any functions within the remaining gradients should be functions of ρ only. As such, Eq. B.4 can be greatly simplified

in our context to $\nabla \times \mathbf{F} = \frac{1}{\rho} \frac{\partial(\rho F_\phi)}{\partial \rho} \mathbf{e}_z$. This is akin to finding $\mathbf{F} = F_\phi \mathbf{e}_\phi$, where $F_\phi = f(\rho)$ to satisfy the dependence on ρ only. Combined with Eq. B.3 we find

$$W(\rho) = \frac{1}{\rho} \frac{\partial(\rho F_\phi)}{\partial \rho} \quad (\text{B.5})$$

Here the steps clearly explain in Sec. 4.1.1 can be followed. The first is to rearrange the equation, then integrate both sides with respect to ρ . An auxiliary variable should be introduced to clarify a distinction between the variable of integration $d\rho$ and the variable ρ outside the integration. In this case, s is used. For the integration, usually $s \in [0, \infty)$, but in this case, ρ is almost always bounded on the boundary of the area, and so we can safely assume for simplicity $s \in [0, \rho]$. This provides our limits of integration. Finally, to normalise the integration limits to $[0, 1]$, a change of variable $s = \rho \hat{s}$ is made. This leads to the key equation

$$F_\phi = f(\rho) = \rho \int_0^1 \hat{s} W(\rho \hat{s}) d\hat{s} \quad (\text{B.6})$$

This allows us to define $\mathbf{F}$ clearly, and can thus apply Stokes' theorem to B.1 directly, utilising the appropriate equivalences discussed. It can be seen that

$$\begin{aligned} \gamma_a &= \oint_{\partial \Sigma} \mathbf{F} \cdot d\mathbf{\Sigma} \\ &= \int_{\partial \Omega} F_\phi \mathbf{e}_\phi \cdot d\mathbf{r}' \end{aligned}$$

At this stage, we can return to the use $\mathbf{r}$ in place of $\mathbf{r}'$ for the integration variable for simplicity of notation.

Whilst this equation is valid as is, there are some simplifications that can be made to improve the consistency of the formulation and the ease of implementation. Firstly, $\mathbf{e}_\phi$ should be rewritten in terms of the vector $\mathbf{r}$ to allow the integration to be evaluated more directly. This can be done by recognising that in cylindrical coordinates, $\mathbf{e}_\phi$ is a 90° anti-clockwise rotation from $\mathbf{e}_\rho$. The 90° anti-clockwise rotation matrix in 2D can be simply verified to be

$$\mathbf{P} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (\text{B.7})$$

and so $\mathbf{e}_\phi = \mathbf{P} \mathbf{e}_\rho$. Further, note that the definition of the unit vector means that $\mathbf{e}_\rho = \frac{\mathbf{r} - \mathbf{r}_a}{\rho}$. These can be combined to give

$$\mathbf{e}_\phi = \frac{1}{\rho} \mathbf{P}(\mathbf{r} - \mathbf{r}_a) \quad (\text{B.8})$$

and

$$\gamma_a = \int_{\partial\Omega} F_\phi \frac{1}{\rho} \mathbf{P}(\mathbf{r} - \mathbf{r}_a) \cdot d\mathbf{r} \quad (\text{B.9})$$

As discussed in Sec. 4.1.1, the final step is to introduce an arbitrary parameterisation for $\mathbf{r} = \mathbf{r}(t)$, where $t \in [t_0, t_1]$. This also acts to simplify the evaluation of the integral by clearly defining limits for the integration. Of course, in practice these will depend on the parameterisation employed, which is addressed in later sections throughout Chapter 4. Using $\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}$, and $d\mathbf{r} = \dot{\mathbf{r}}dt$, it can finally be written that

$$\gamma_a = \int_{t_0}^{t_1} F_\phi \frac{1}{\rho} \mathbf{P}(\mathbf{r} - \mathbf{r}_a) \cdot \dot{\mathbf{r}} dt \quad (\text{B.10})$$

which returns Eq. (76a) in combination with the definition of F_ϕ in Eq. B.6. The 2D area integral for γ is thus fully transformed to a 1D line integral through the use of Stokes' theorem. It also noted that $\mathbf{P}(\mathbf{r} - \mathbf{r}_a) \cdot \dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^T \mathbf{P}(\mathbf{r} - \mathbf{r}_a)$.