

Modelling of Fluid Sloshing using Smoothed Particle Hydrodynamics

Enson Tin Hang Un

CID: 01355016

Department of Aeronautics

Imperial College London

Supervisor: Prof Joaquim Peiró

Submitted in partial fulfilment of the requirements for the Degree of
Master of Engineering in Aeronautical Engineering

1 June 2021

Abstract

This report presents the formulation of a high-order, low-dissipation Riemann solver based Smoothed Particle Hydrodynamics (SPH) method for modelling free-surface flows. The method aims to generate stable and accurate solutions automatically, without user-input parameters. Two novel techniques are proposed to facilitate the method, including a Taylor-series based stencil construction method and an artificial diffusion limiter. High-order schemes, such as Weighted Essentially Non-oscillatory (WENO) reconstruction, are also implemented to raise the order of accuracy. Numerical tests show that our method produces smooth and accurate solutions with significantly less numerical damping than the traditional method. Moreover, we show that the higher-order schemes consistently achieve a higher order of accuracy and improved computational efficiency over the low-order methods. Directions for potential improvement are discussed.

Acknowledgements

First of all, I would also like to thank my family for their unconditional love and support throughout my education. That is why I can study here and submit this report.

I would like to thank my FYP supervisor, Professor Joaquim Peiró, for his advice and guidance during the project, particularly for his patience in answering all my questions. I will never forget his teaching of ‘do not say no until you’re told not to’.

I would like to extend my gratitude to my personal tutor and UROP supervisor, Dr Chris Cantwell. Thank you for offering me my first research opportunity, and introducing me to Linux, C++, and the world of HPC.

I would also like to thank Cindy for her support and motivation. You are, and will always be, the special one.

Last but not least, I would like to say thank you to my friends, Jason, Jeffrey, Alex, and Kelvin. I have learnt a lot from you guys, and my four years at Imperial would have been unimaginably boring and tedious without your company. All the best in the future.

Contents

1	Introduction	1
1.1	Background and motivation	1
1.2	Godunov-type schemes	2
1.3	Riemann solver based SPH	3
1.4	Aims and objectives	4
1.5	Outline	4
2	Classical SPH formulation	6
2.1	Kernel smoothing approximation	6
2.1.1	Common choices of kernel function	6
2.2	Discretised SPH operators	8
2.2.1	Interpolation	8
2.2.2	Gradient, divergence, and curl	8
2.3	Governing Equations	9
2.3.1	Compressible, incompressible, and weakly-compressible flow assumptions	9
2.3.2	Discretising the continuity equation	10
2.3.3	Discretising the momentum equation	10
2.4	On error analysis in SPH	11
2.5	Boundary treatment	12
2.6	Time integration scheme	14
3	Riemann solver based SPH	15
3.1	Modified equations of motion	15
3.2	Constructing stencils	15
3.2.1	Taylor series based extrapolation	15
3.2.2	Advanced derivatives	16
3.3	Reconstruction schemes	19
3.3.1	Baseline / Godunov's scheme	19

3.3.2	Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL)	19
3.3.3	Weighted Essentially Non-oscillatory (WENO) scheme	20
3.4	Riemann solver	20
3.5	Modified boundary treatment	21
4	Implementation	22
5	Numerical tests	22
5.1	Hydrostatic case	22
5.1.1	Artificial diffusion limiter	23
5.2	Dam-breaking	26
5.2.1	Comparing to analytical and numerical works	26
5.2.2	Comparing to experimental data	29
5.3	Oscillating droplet in a conservative force field	30
5.3.1	Convergence study	31
5.3.2	Computational efficiency	33
5.4	Linear sloshing in a rectangular tank	34
6	Conclusions	37
6.1	Future works	38
References		39
A	Lid-driven cavity: An example of SPH viscosity model and moving wall boundary condition	45

List of Tables

1	Comparison between different boundary treatment methods	13
2	Comparison between different discrete first derivative operators	16
3	Different discrete second derivative operators	17
4	Parameters in the hydrostatic case	22
5	Parameters in the dam-breaking case (Unspecified parameters remain the same as the previous case.)	26
6	Errors and order of convergence in the finest range.	32

List of Figures

1	SPH interpolation	1
2	Schematics of the Godunov' scheme (modified from [78])	2
3	W_3 , W_4 , W_5 , and their first and second derivatives normalised	7
4	Frequency spectra of compressible, weakly-compressible, and incompressible flows. (Light grey: purely incompressible features; Dark grey: acoustic part.) [85]	9
5	SPH convergence behaviour in (a,b) regular particles and (c) disordered particles with zeroth-order consistent scheme.	12
6	Different boundary treatment approaches: (a) mirror particles; (b) fixed particles; (c) repulsive particles; and (d) kernel renormalisation. Shaded area indicates the area of the kernel in normalisation.	12
7	Schematics of the four-point stencil used in this work	15
8	Convergence study of first and second derivative schemes with $\Delta x/h = 2/3$ and fifth-order Wendland kernel. (Solid line: regular points; Dashed line: 10% Δx perturbation applied.)	17
9	A general 1D three-point finite difference stencil	18
10	Different reconstruction schemes: (a) piecewise constant; (b) piecewise linear; and (c) piecewise parabolic (illustrated the reconstruction for left state only).	19
11	Setup of the hydrostatic case	22
12	Energy evolution of δ -SPH and different reconstruction schemes. (Notes: δ -SPH uses $\alpha = 0.24$, $\delta = 0.02$ as suggested by [24]; Second derivative is switched on for stencil extrapolation in WENO-Z; and $t_{ref} = \sqrt{g/H}$ is the reference time scale.)	23

15	Near-surface particles see a net negative $p_L - p_R$	23
13	Normalised pressure vs normalised particle height (H is fluid initial height)	24
14	Error field of different schemes at $t/t_{ref} = 100$	25
16	Setup of the dam-breaking case at $t = 0$	26
17	Energy evolution of δ -SPH ($\alpha = 0.1$, $\delta = 0.02$) and Riemann solver SPH with different reconstruction schemes in the dam-breaking case	26
18	Pressure field of different schemes at $t/t_{ref} = 6.1$	27
19	Snapshots of the normalised pressure field in dam-breaking case (Leftmost: Adami et al. [1])	28
20	Wavefront location of different schemes against analytical result [71]	29
21	Pressure evolution at $\mathbf{x}_p = [5.36, 0.2]^\top$ of different schemes in dam-breaking case comparing to the experimental data from Buchner [8] (black triangles)	29
22	Setup of the oscillating droplet case. R is the initial radius of the droplet, $a(t)$ and $b(t)$ are the semi-major and semi-minor axis of the evolving ellipse, respectively. For SPH simulations, $v_{max} = 1.5A_0$.	30
23	Snapshots of the oscillating droplet with $\Delta x = 0.05$. All cases show good agreement with the analytical shape (red dashed line), and the pressure fields are smooth.	31
24	Searching for a sufficiently small $\Delta x/h$	31
25	Convergence of different schemes in oscillating droplet case ($h/\Delta x = 1.5$)	32
26	Runtime of different schemes against particle resolution	33
27	Setup of the linear sloshing case	34
28	Surface elevation at the left wall ($x = 0$)	35
29	Fourier transform of the surface elevation at the left wall	36
30	Snapshots of different schemes in linear sloshing case. Red dashed line: analytical solution $H(x, t)$.	36
31	Setup of the lid-driven cavity case	45
32	x -velocity along $x = 0.5$ and y -velocity along $y = 0.5$ at steady state	46
33	Velocity magnitude and pressure distribution at steady state	46

1 Introduction

1.1 Background and motivation

Smoothed Particle Hydrodynamics (SPH), originally formulated by Gingold and Monaghan [32] and Lucy [49] in 1977, is a completely meshless method for simulating continuum mechanics. It represents fields (e.g. density, velocity, pressure) and their derivatives using a set of particles and a smoothing kernel function. For example, the density at a point in space is computed as a weighted sum of the density of all particles around that point. Particles closer to the point of interest contribute more to the summation, *vice versa*. This concept of varying ‘relevance’ with distance is handled by the smoothing kernel function. It is essentially the same as applying a spatial smoothing filter to the particles, hence the name.

Comparing to the mesh-based CFD methods such as Finite Difference (FD), Finite Volume (FV), and Level-set Method (LSM), the meshless nature of SPH makes it more suitable for free-surface, multi-phase flows, Magnetohydrodynamics (MHD) [63, 64], as well as solids with large deformation and fracture [39, 62]. Despite the attractive capabilities, SPH suffers from various numerical issues. In particular, this project concerns the issues of numerical instability and low order of accuracy.

Firstly, since the SPH discretisation (see Section 2.3) resembles the central differencing in FD, it produces spurious oscillations with hyperbolic conservation laws [51]. Various forms of artificial viscosity and artificial diffusion (collectively referred to as artificial dissipation) have been introduced to stabilise the solution. Most commonly used ones are the artificial viscosity model by Monaghan and Gingold [55] and the more recent δ -SPH by Antuono *et al.* [2]. Nevertheless, all of these formulations rely on user-input parameters to control the strength of dissipation. The parameters are dependent on the problem, which require timely tuning such that the loss in energy remains low but sufficient for stabilising the solution. The formulations of SPH that require such explicit dissipation parameters are hereafter referred to as *classical* SPH.

Secondly, although Monaghan [54] and Quinlan *et al.* [67] showed that, in theory, classical SPH with Gaussian-like kernel functions can achieve second-order convergence, in practice, SPH codes usually have convergence order only between 1.1 and 1.8 [29, 30, 45, 81]—significantly lower than its mesh-based counterparts. The order of accuracy is limited by the accuracy of the interactions between particles in solving the discretised governing equations. Therefore, high-order schemes for evaluating particle interactions are desirable for increasing computational efficiency.

To address the above issues, Inutsuka [41], Vila [84], and Ben Moussa [4] formulated a new class of SPH, which uses the *Godunov-type scheme* to calculate particle interactions. In the following

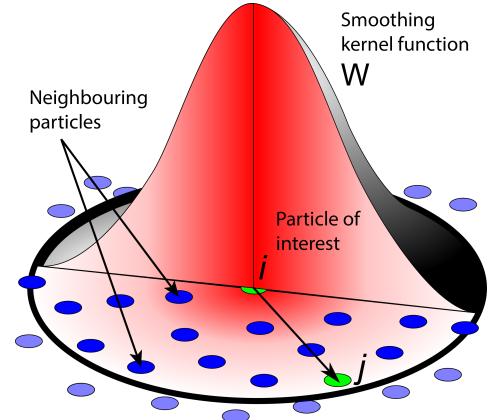


Figure 1: SPH interpolation

sections, we will introduce the Godunov-type schemes that originated in FV, its adaptation to SPH, and a brief history of the development of Riemann solver based SPH.

1.2 Godunov-type schemes

A Riemann problem is an initial value problem with piecewise constant initial conditions joined by a step discontinuity at an interface. In 1959, Godunov [33] pioneered the idea of using a local Riemann problem to solve conservation laws in FV. He treated each FV cell to have a constant solution equal to the cell average. At the interface, there are two piecewise constant states (referred to as the left and the right state) joined by a discontinuity—a Riemann problem locally. The solution at the next timestep can be obtained by solving the local Riemann problem, then using the value at the interface to compute numerical fluxes, and integrate in time.

Godunov’s method, because of the piecewise constant representation, is first-order accurate. Van Leer’s Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) [82] extended the method by replacing the piecewise constant representation with a linear one, making the scheme second-order accurate. The linear function on each side of the interface is constructed using the cell next to the interface and its adjacent cell, i.e. four cells are needed to evaluate the left and right states in this scheme. This group of cells is named a *stencil*. Van Leer’s scheme opened up the idea of using higher-order polynomials to extrapolate for the states of the local Riemann problem, so to increase accuracy. This procedure is known as *reconstruction*.

However, higher-order polynomials/schemes produce spurious oscillations (or wiggles) in the vicinity of discontinuities or shocks [37]. Harten [35] suggested that the solution will be oscillation-free if the numerical scheme does not generate new maximum or minimum in the solution in every timestep—the concept known as Total Variation Diminishing (TVD). Practically, a flux/slope limiter is employed, which returns the reconstruction to first-order if it violates the TVD condition. Sweby [77] proved the necessary and sufficient conditions for second-order TVD limiters, giving rise to a range of limiters, each having different characteristics but all capable of enforcing the TVD condition. Some applied quadratic reconstruction with the second-order limiters, but there is no guarantee that the solution is TVD.

A big leap towards arbitrarily high-order reconstruction is the Essentially Non-oscillatory (ENO) scheme introduced by Harten *et al.* [36], in 1987. The scheme divides the stencil into a number of sub-stencils, then construct a polynomial on each sub-stencil. The polynomials must preserve the cell average in each cell for conservation properties. The key step is to select the ‘smootherest’

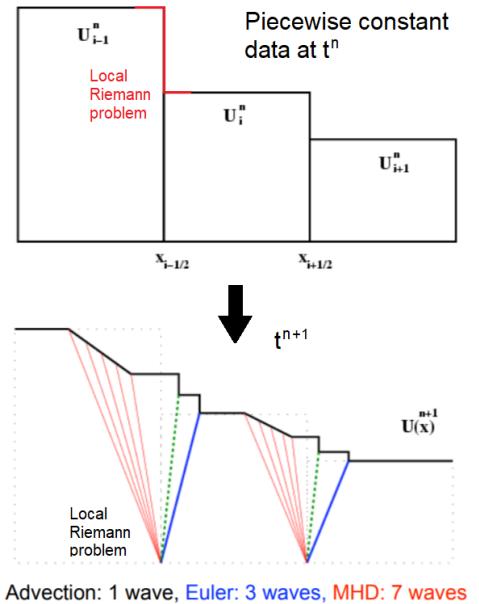


Figure 2: Schematics of the Godunov scheme (modified from [78])

making the scheme second-order accurate using the cell next to the interface and its adjacent cell, i.e. four cells are needed to evaluate the left and right states in this scheme. This group of cells is named a *stencil*. Van Leer’s scheme opened up the idea of using higher-order polynomials to extrapolate for the states of the local Riemann problem, so to increase accuracy. This procedure is known as *reconstruction*.

However, higher-order polynomials/schemes produce spurious oscillations (or wiggles) in the vicinity of discontinuities or shocks [37]. Harten [35] suggested that the solution will be oscillation-free if the numerical scheme does not generate new maximum or minimum in the solution in every timestep—the concept known as Total Variation Diminishing (TVD). Practically, a flux/slope limiter is employed, which returns the reconstruction to first-order if it violates the TVD condition. Sweby [77] proved the necessary and sufficient conditions for second-order TVD limiters, giving rise to a range of limiters, each having different characteristics but all capable of enforcing the TVD condition. Some applied quadratic reconstruction with the second-order limiters, but there is no guarantee that the solution is TVD.

A big leap towards arbitrarily high-order reconstruction is the Essentially Non-oscillatory (ENO) scheme introduced by Harten *et al.* [36], in 1987. The scheme divides the stencil into a number of sub-stencils, then construct a polynomial on each sub-stencil. The polynomials must preserve the cell average in each cell for conservation properties. The key step is to select the ‘smootherest’

polynomial and use it to extrapolate to the interface, avoiding discontinuities being brought into the calculation.

Improving on the ENO idea, Liu, Osher and Chan [48] developed the Weighted Essentially Non-oscillatory (WENO) scheme that achieves higher order of accuracy with the same stencil size—an n th-order ENO can be modified to an $(n + 1)$ th-order WENO. WENO computes the interface value as a weighted sum of all polynomials. Therefore, even the information of the non-smooth polynomials are utilised. The weights are calculated based on the smoothness of the polynomials. A few modifications were made to the original WENO. For example, the WENO-JS by Jiang and Shu reformulated the polynomial smoothness indicators to yield $(2n - 1)$ th-order accuracy in some cases. The latest development is the WENO-Z by Borges *et al.* [7] and Castro *et al.* [9], which took into account smoothness at both local (sub-stencil) and global (overall stencil) scales. The new scheme displayed significant improvement in accuracy in both discontinuous and smooth regions.

After reconstructing the local Riemann problem, the solution at the interface, which is constant over time, can be obtained. This is done either iteratively using an *exact* Riemann solver, or using an *approximate* Riemann solver for a more manageable computational cost. A complete discussion on Riemann solvers can be found in Ref. [79].

1.3 Riemann solver based SPH

A Riemann solver based SPH, also known as a Godunov-type SPH (GSPH), computes fluxes between particles using the Godunov-type scheme from FV. Since there is no ‘cell’ or ‘stencil’ structure in SPH, the first step of a Riemann solver based SPH is to construct an FV-like stencil. It is then followed by the same procedures as in the FV Godunov-type schemes: reconstruction of the left and right states of the Riemann problem, and passing them to a Riemann solver.

As early as 1994, Inutsuka [41] implemented a GSPH using the original Godunov’s scheme. Two imaginary points, one ahead of and one behind the particle of interest, are constructed along the direction of the pressure gradient, using SPH interpolation. The method was shown to be successful in astrophysical flows with strong shocks. Inutsuka [42] later reformulated the method to using MUSCL reconstruction with van Leer’s slope limiter [82]. A 1D 4-point stencil, along the line joining the interacting particles, is used. Yet, the exact Riemann solver remained.

Parshikov and Medin [62] moved away from exact Riemann solvers and opted for Roe’s approximate Riemann solver [73]. Their code simulated compressible gas and impact of elastic materials and was reported to achieve lower dissipation and better accuracy than the classical SPH formulation in [15].

In 2014, Avesani *et al.* [3] introduced the first WENO-SPH, which uses the polynomial WENO method developed by Friedrich [31], Käser and Iske [44], and Dumbser *et al.* [23] instead of the traditional 1D WENO introduced in the previous section. Around the interested particle, nine 2D polynomials are constructed using the Moving Least Square (MLS) approach. One polynomial is in the centre region and eight in the neighbouring sectors, covering all possible

directions of interaction. Each time, one of the eight neighbouring polynomials will be activated depending on the angle formed by the interacting particle pair. The left and right states of the Riemann problem were obtained by a weighted sum of the centre polynomial and the activated polynomial evaluated at the midpoint of the interacting particle pair. The weights are determined using WENO scheme. Rusanov [74] and Osher [25, 61] approximate Riemann solvers were then applied to solve for the fluxes for compressible Euler equations. The scheme demonstrated a fourth-order accuracy when cubic polynomials were constructed. Although this method displays good accuracy, it is very computationally expensive.

Recently, Zhang [91] suggested an alternative WENO method for Weakly-compressible SPH (WCSPH). He returned to the traditional 1D WENO reconstruction using the 4-point stencil along the direction of the interacting particle pair. The values at the two imaginary particles are extrapolated from the interacting particles using the gradients at the particles. This method is simple and fast, but the linear extrapolation is suspected to have limited the order of accuracy of the WENO scheme. Also, it was claimed that the WENO is fifth-order accurate, which is not documented in literatures for 4-point WENO schemes. Nevertheless, with Roe's solver and a new dissipation limiter [90], the method demonstrated low numerical dissipation with smooth pressure and density fields. As reported by Hu *et al.* [40], the same stencil construction method also works well with MUSCL minmod limiter [72] in simulating compressible multi-phase flows.

1.4 Aims and objectives

Although Riemann solver based SPH is not a new idea, higher-order, especially WENO, SPH is a relatively new area with only a few (less than 10) publications available. This project aims to assess the feasibility, suitability and performance of high-order Riemann solver based SPH techniques in simulating free-surface flow problems, such as sloshing. We depart from Zhang's works [88, 90, 91], because (i) it is the only WENO-SPH scheme developed and tested for hydrodynamic applications, and (ii) it is simple and computationally affordable, which is a key consideration in this project.

The specific objectives are

1. To implement, verify, and validate a Riemann solver based SPH, and to confirm its implicit numerical dissipation against classical SPH simulations and experimental data;
2. To derive and study a high-order stencil extrapolation method, so that the extrapolation order matches the reconstruction order; and
3. Based on the newly developed stencil construction method, to implement the second-order MUSCL and third-order WENO reconstruction schemes, and to test if they achieve the promised rate of convergence.

1.5 Outline

This report is structured as follows.

Chapter 2 will present the mathematical foundations of SPH methods. We will demonstrate how a scalar field is approximated and discretised using kernel and particles. The SPH representation will then be used to discretise the governing equations. A short summary of the convergence behaviour of SPH is given, followed by details on treatments to impose boundary conditions and the time integration scheme. **Chapter 3** will go through, step-by-step, how a classical SPH is reformulated into a Riemann solver SPH. We will introduce our stencil extrapolation method and study its accuracy. Details of the implementation of different reconstruction schemes will be outlined. And the Riemann solver of our choice, Roe's Riemann solver, will be described. A short note on the code implementation and source code will be in **Chapter 4**. In **Chapter 5**, we will present four numerical examples, each assessing different aspects of the code. We will compare our method against classical SPH simulations and experimental data to examine its stability, accuracy, and dissipation in both static and dynamic conditions. A convergence test will be performed using a benchmark test case, where the order of accuracy and the computational costs of the schemes will be analysed. We will also confirm the ability of our code in simulating sloshing problem. Finally, **Chapter 6** will present our conclusions and some potential directions for future work.

A note on notations: This report uses subscripts and superscripts to denote different quantities and concepts. To avoid confusion, subscripts h, i, j, k are dedicated for fluid particles, and subscripts l, m are always for wall particles. Superscripts denote the timestep number. Other notations will be introduced as they appear.

2 Classical SPH formulation

2.1 Kernel smoothing approximation

SPH centres around approximating an arbitrary scalar function $f(\mathbf{x})$ as

$$\begin{aligned} f(\mathbf{x}_i) &= \int_{\Omega} f(\mathbf{x}_j) \delta(\mathbf{x}_i - \mathbf{x}_j) dV(\mathbf{x}_j) \\ &\approx \int_{\Omega} f(\mathbf{x}_j) W(\mathbf{x}_i - \mathbf{x}_j, h) dV(\mathbf{x}_j) := \langle f(\mathbf{x}_i) \rangle, \end{aligned} \quad (1)$$

where \mathbf{x}_i is the position of particle i , Ω denotes the domain, $\delta(\mathbf{r})$ is the Dirac delta function, and $W(\mathbf{r}, h)$ is the kernel function characterised by a parameter named smoothing length, h .

For simplicity, from here onwards, we will write

$$f_i := f(\mathbf{x}_i), \quad W_{ij} := W(\mathbf{x}_i - \mathbf{x}_j, h), \quad \mathbf{r}_{ij} := \mathbf{x}_i - \mathbf{x}_j, \quad r_{ij} := \|\mathbf{r}_{ij}\|, \quad V_j := V(\mathbf{x}_j).$$

One way of understanding this approximation is that the Dirac delta function $\delta(\mathbf{r})$ in the convolution integral is approximated by the kernel function $W(\mathbf{r}, h)$. In order to guarantee convergence as $h \rightarrow 0$, $W(\mathbf{r}, h)$ must satisfy

$$\int_{\Omega} W(\mathbf{r}, h) dV = 1, \quad (2)$$

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

and

$$\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r}). \quad (4)$$

Apart from the above necessary conditions, Price [64] and Wang *et al.* [86] suggested other properties that a good interpolation kernel should have:

1. Symmetric/Even: $W(\mathbf{r}, h) = W(-\mathbf{r}, h) = W(r, h)$, where $r \geq 0$;
2. Always positive: $W(r, h) \geq 0 \quad \forall r \in \mathbb{R}^+$;
3. Decreases monotonically with r : $W(r_1, h) \geq W(r_2, h) \quad \forall r_2 \geq r_1$;
4. Has sufficiently smooth derivatives;
5. Has sufficiently high-order derivatives equal to zero at the boundary (also known as boundary smoothness); and
6. Has a flat central portion so the interpolation is resistant to perturbations of neighbouring particles' position.

2.1.1 Common choices of kernel function

Early SPH [32] used Gaussian as kernel function as it naturally satisfies all the above conditions.

$$W(r, h) = \frac{\sigma}{h^d} \exp\left(\frac{-r^2}{h}\right), \quad (5)$$

where $\sigma = \pi^{-d/2}$ is the normalisation factor in d -dimension to satisfy Eq.2. Although Gaussian is one of the most accurate kernel functions for interpolation in smooth fields [38], its infinite support, i.e. $W(r, h) > 0 \quad \forall r \in \mathbb{R}^+$, makes it inefficient as every particle needs to interact with all other particles in the domain—an $\mathcal{O}(N^2)$ operation. In theory, some [67] argued that infinite support correctly models incompressible flow, as information is non-local. In practice, for numerical efficiency, Gaussian-like shaped kernels with *compact support* are preferred.

It is important to emphasise that there is no agreement on the ‘best’ kernel function so far. Yet, there are many proposed options that can adequately serve the purpose. The most commonly-used kernels are listed in below, readers can refer to Wang *et al.* [86] for more such choices.

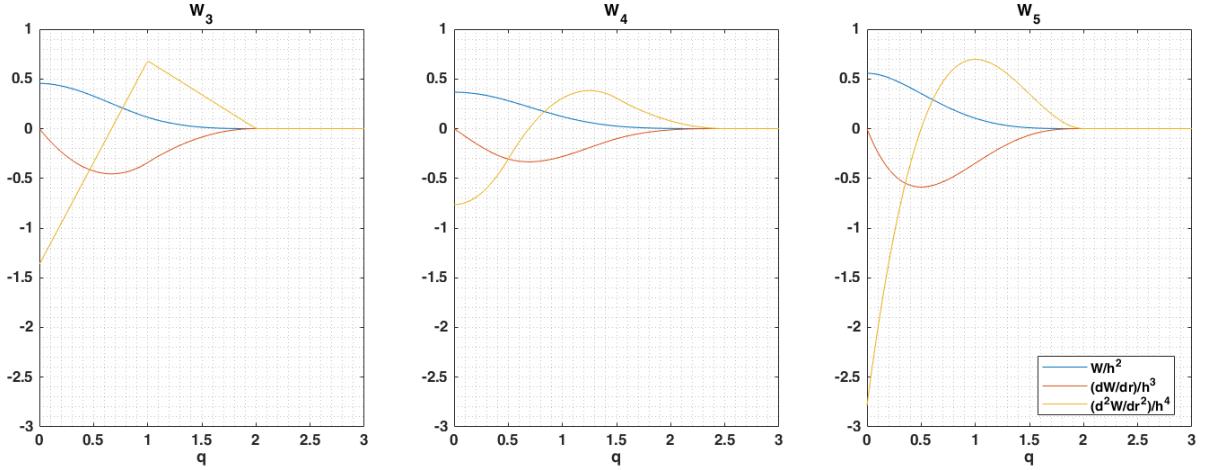


Figure 3: W_3 , W_4 , W_5 , and their first and second derivatives normalised

Cubic and quartic B-spline: This family of B-spline kernels, introduced by Monaghan and Lattanzio [57], is one of the earliest compact-support kernels in SPH. They are commonly used mainly because of their narrow support radius and low computational cost. On the downside, they suffer from tensile instability—a condition that leads to particle clumping and formation of voids in negative pressure zones.

$$W_3(r, h) = \frac{\sigma}{h^d} \begin{cases} \frac{1}{4}(2-q)^3 - (1-q)^3, & 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3, & 1 \leq q < 2 \\ 0, & q \geq 2 \end{cases} \quad (6)$$

where $q = r/h$ and $\sigma = [2/3, 10/(7\pi), 1/\pi]$ for [1,2,3]-dimension respectively. The *support radius* of a kernel is expressed as kh , where k is a positive constant, for instance, in this case, $k = 2$.

$$W_4(r, h) = \frac{\sigma}{h^d} \begin{cases} \left(\frac{5}{2}-q\right)^4 - 5\left(\frac{3}{2}-q\right)^4 + 10\left(\frac{1}{2}-q\right)^4, & 0 \leq q < \frac{1}{2} \\ \left(\frac{5}{2}-q\right)^4 - 5\left(\frac{3}{2}-q\right)^4, & \frac{1}{2} \leq q < \frac{3}{2} \\ \left(\frac{5}{2}-q\right)^4, & \frac{3}{2} \leq q < \frac{5}{2} \\ 0, & q \geq \frac{5}{2} \end{cases} \quad (7)$$

$\sigma = [1/24, 96/(1199\pi), 1/(20\pi)]$ for [1,2,3]-dimension respectively; $k = 2.5$.

Quintic Wendland: Wendland [87] developed this family of kernel functions to combat the issue of tensile instability. All of his kernels have non-negative Fourier transform, which is a

necessary condition for stability according to Dehnen [20]. Among all, the fifth-order Wendland kernel is the most popular as it has a good balance between complexity and smoothness.

$$W_5(r, h) = \frac{\sigma}{h^d} \begin{cases} (1+2q)(2-q)^4, & 0 \leq q < 2 \\ 0, & q \geq 2 \end{cases}, \quad (8)$$

$\sigma = [3/64, 7/(64\pi), 21/(256\pi)]$ for $d = [1, 2, 3]$ respectively; $k = 2$.

2.2 Discretised SPH operators

2.2.1 Interpolation

If the previous section explains why SPH is named ‘smoothed’, now we will discuss the elements of ‘particle’. Analogous to the Riemann sum, the integral in Eq.1 is approximated by a particle-wise summation, i.e.

$$\langle f_i \rangle \approx \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} f_j W_{ij} := \llbracket f_i \rrbracket, \quad (9)$$

where $m_j/\rho_j = V_j$ is the volume of particle j , and \mathcal{P}_i denotes the set containing all particles in the *neighbourhood* of i , i.e. $\mathcal{P}_i = \{j \mid r_{ij} \leq kh\}$.

2.2.2 Gradient, divergence, and curl

We first consider the gradient operator. Starting from Eq.1, using integration by part,

$$\langle \nabla_i f_i \rangle = \int_{\Omega} (\nabla_j f_j) W_{ij} dV_j = \int_{\Omega} \nabla_j (f_j W_{ij}) dV_j - \int_{\Omega} f_j (\nabla_j W_{ij}) dV_j, \quad (10)$$

where ∇_i denotes gradient with respect to \mathbf{x}_i .

The gradient theorem states that $\oint_{\partial\Omega} \omega = \int_{\Omega} d\omega$, so the first term on the right-hand-side becomes

$$\oint_{\partial\Omega} f_j W_{ij} dS_j. \quad (11)$$

$\partial\Omega$ denotes the boundary of Ω . The above expression is strictly zero because W vanishes at the compact support radius. This is also true for infinite support kernels using Eq.3. Next, assume the kernel is symmetric (Condition 1 in Section 2.1), we can apply a change of variables,

$$\nabla_j W_{ij} = \nabla_i W_{ji} = -\nabla_i W_{ij}. \quad (12)$$

Eq.10 finally becomes

$$\langle \nabla_i f_i \rangle = \int_{\Omega} f_j (\nabla_i W_{ij}) dV_j \approx \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} f_j \nabla W_{ij} := \llbracket \nabla_i f_i \rrbracket. \quad (13)$$

We will neglect the subscript of ∇ hereafter. For axis-symmetric kernels,

$$\nabla W_{ij} = \frac{dW}{dr} \mathbf{e}_{ij}, \quad (14)$$

where $\mathbf{e}_{ij} = \mathbf{r}_{ij}/\|\mathbf{r}_{ij}\|$.

Similarly, the divergence and curl operators can be obtained by applying Gauss's divergence theorem and Stokes' theorem respectively.

$$[\nabla \cdot \mathbf{f}_i] = \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} \mathbf{f}_j \cdot \nabla W_{ij}, \quad (15)$$

$$[\nabla \times \mathbf{f}_i] = \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} \mathbf{f}_j \times \nabla W_{ij}, \quad (16)$$

These are the *basic* first derivative operators. However, they are never used in practice. Using Taylor series expansion (see e.g. [27, 67]), it can be shown that the truncation error of these operators are proportional to f_i , making them zeroth-order accurate. Nevertheless, they remain the fundamental building blocks of more advanced derivative schemes which will be introduced in later sections.

2.3 Governing Equations

The Euler equations in Lagrangian frame of reference read

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{u} \quad (17a)$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \mathbf{g} \quad (17b)$$

where $\frac{D(\bullet)}{Dt} = \frac{\partial(\bullet)}{\partial t} + \mathbf{u} \cdot \nabla(\bullet)$ denotes the total (i.e. material) derivative. The system of equations is closed using the *weakly-compressible* condition, which will be discussed below.

2.3.1 Compressible, incompressible, and weakly-compressible flow assumptions

The above system is *unclosed* as it contains three unknowns (ρ, \mathbf{u}, p) but only two equations. The difference between different flow assumptions lies in the choice of the third equation to close the system.

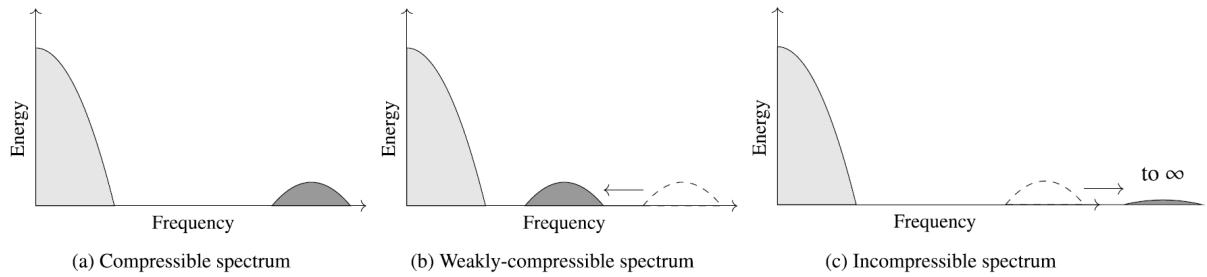


Figure 4: Frequency spectra of compressible, weakly-compressible, and incompressible flows. (Light grey: purely incompressible features; Dark grey: acoustic part.) [85]

Compressible flow carries no assumption, as all flows are compressible in reality. The system is closed by an energy conservation equation and the ideal gas law. However, when pressure variation is small, e.g. at low Mach numbers, numerical error is comparable to the solution, resulting in poor accuracy [85]. Also, this formulation allows acoustic waves (infinitesimal pressure waves) to travel at the physical speed of sound, 1480m/s for water at room temperature and pressure, which will require very fine timesteps, and therefore, high computational cost.

The incompressible flow model assumes that density is constant throughout the fluid. Pressure is sought through a Poisson equation [13]

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right)_i = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}_i^*. \quad (18)$$

The key feature of a Poisson equation is that the solution at one point depends on the solution over the entire domain. Despite loosening the timestep constraint, incompressible schemes need to solve a large matrix system, which is expensive in both memory and computation.

The weakly-compressible (WC) assumption strikes the middle ground of the two. It links pressure and density using an equation of state (EoS). The most commonly used barotropic EoS is

$$p = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right]. \quad (19)$$

In this formulation, acoustic waves travel at an artificial speed of sound c_0 , which is selected such that it is sufficiently higher than the incompressible flow features, but remain lower than the physical speed of sound for numerical efficiency. A common choice for SPH is $c_0 = 10 v_{\max}$, resulting in a Mach number of about 0.1 and a density variation $|\rho/\rho_0| < 1\%$. In practice, v_{\max} can be analytically or empirically determined. Moreover, γ is a constant dependent on the fluid, usually taken as 7 for water and 1.4 for air (it is *not* the heat capacity ratio). Since $|\rho/\rho_0|$ is small anyway, some WCSPH (e.g. [70, 88]) use $\gamma = 1$ for all fluids for simplicity.

2.3.2 Discretising the continuity equation

Since we keep the mass of all particles constant throughout the simulation, conservation of mass is naturally satisfied. Monaghan [54] suggested a trick to improve accuracy of the basic SPH operators, which is by the chain rule expression

$$\varphi \nabla \cdot \mathbf{u} = \nabla \cdot (\varphi \mathbf{u}) - \mathbf{u} \cdot \nabla \varphi, \quad (20)$$

where φ can be any differentiable function. If we choose $\varphi = 1$, and discretise $\nabla \cdot \mathbf{u} \rightarrow [\![\nabla \cdot \mathbf{u}_i]\!]$,

$$[\![\nabla \cdot \mathbf{u}_i]\!]_{\text{new}} = [\![\nabla \cdot \mathbf{u}_i]\!]_{\text{basic}} - \mathbf{u}_i \cdot [\![\nabla 1]\!]. \quad (21)$$

Substituting into Eq.17a yields

$$\frac{D\rho_i}{Dt} = \rho_i \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} (\mathbf{u}_i - \mathbf{u}_j) \cdot \nabla W_{ij}. \quad (22)$$

This formulation is widely used and tested robust in violent flow simulations. It is zeroth-order consistent, which means that the first derivative of a constant function is exactly zero. Also, the $(\mathbf{u}_i - \mathbf{u}_j)$ term is why people say SPH resembles the central difference in FD.

2.3.3 Discretising the momentum equation

It is important for long term stability and accuracy that the numerical scheme conserves linear and angular momentum exactly. Consider $\frac{D\mathbf{u}_i}{Dt} = \sum_j \mathbf{F}_{ij}$ is the semi-discretised momentum

equation, a discretisation is called *symmetric* if and only if $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$, i.e. it satisfies Newton's third law. One of such schemes can be obtained similar to Eq.21, with new and basic schemes swapped:

$$[\nabla p_i]_{\text{basic}} = [\nabla p_i]_{\text{new}} - p_i [\nabla 1] \implies [\nabla p_i]_{\text{new}} = \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} (p_i + p_j) \nabla W_{ij}. \quad (23)$$

Hence, substituting into Eq.17b yields

$$\frac{D\mathbf{u}_i}{Dt} = -\frac{1}{\rho_i} \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} (p_i + p_j) \nabla W_{ij} + \mathbf{g}. \quad (24)$$

The same expression can be derived from Hamilton's variational principle and the Euler-Lagrange equation, which proves the origin of its conservation property. Interested readers can find the derivation in almost all PhD thesis on SPH (e.g. [34, 63, 88]). Note that this discretisation is not zeroth-order consistent, which may hinder its convergence. However, the long term conservation property outweighs this disadvantage.

Remarks on viscosity: Viscosity is neglected in this formulation. This is because SPH viscosity is a complicated topic, which, itself is worth becoming another research project. Also, this project focuses on numerical dissipation of SPH schemes. By removing physical viscosity, all losses in energy will be solely due to the numerical dissipation of the scheme, making investigation easier. Nevertheless, interested readers can have a look at Appendix A for an example of how viscosity is modelled in SPH.

2.4 On error analysis in SPH

The error associated with SPH is different from that in mesh-based schemes. There are two sources of error in SPH: the *kernel approximation* and the *particle discretisation*. Although a detailed investigation on SPH error is beyond the scope of this report, having some insights into the characteristics of SPH error will be helpful when assessing the accuracy of our scheme. Below are the key conclusions from Monaghan [54], Quinlan [67], and Fatechi [27]:

Kernel approximation error: For symmetric positive kernels away from boundaries (so it has full support), the truncation error of the kernel approximation (Eq.1) is $\mathcal{O}(\varsigma h^2 + h^4)$, where ς is a constant depending on the kernel. Monaghan [52] suggested a family of kernels with $\varsigma = 0$. Despite the appealing fourth-order accuracy, they lead to other problems and are rarely used. At present, there is no known zero- ς kernel that does not induce other side effects. Note that this error is independent of particle distribution. Also, the order of accuracy in kernel approximation may not be equal to the order of accuracy of the overall SPH code, as, in practice, other sources of error may be dominating, such as particle interaction, and the lack of full kernel support.

Particle summation error: For equi-spaced particles, the error of particle discretisation is $\mathcal{O}(\Delta x/h)^{b+2}$, where b is the kernel boundary smoothness. It is defined as the integer such that the b -th derivative and all lower derivatives of the kernel are zero at the boundary of the compact support (b is not well-defined for infinite support kernels). Therefore, SPH convergence, different from mesh-based method, requires both $h \rightarrow 0$ and $\Delta x/h \rightarrow 0$. When speaking about the 'order

of convergence' of an SPH scheme in a mesh-based method sense, one should always be careful that $\Delta x/h$ must be sufficiently small for the convergence curve to reach the asymptote.

When particles are disordered: The error is proportional to h^2 when h is sufficiently large, but the error becomes of order h^{-1} when h gets smaller. However, first-order consistent schemes can remove this divergent behaviour. The order of consistency is defined as the highest order of the polynomial which the scheme can calculate its derivative exactly. The more disordered the particles are, the faster the error departs from h^2 , i.e. a larger critical h . The order of convergence on $\Delta x/h$ is more complicated. It depends on h , b , particle disorderness, and the numerical scheme.

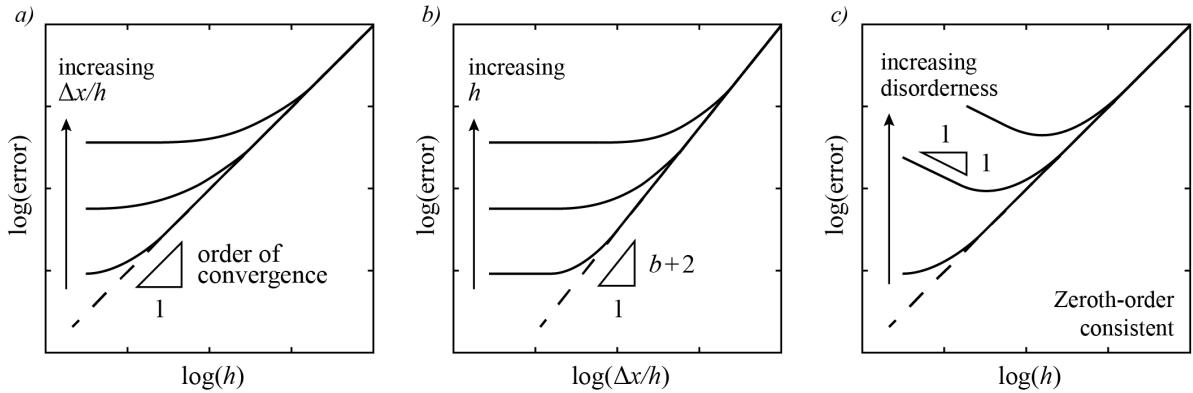


Figure 5: SPH convergence behaviour in (a,b) regular particles and (c) disordered particles with zeroth-order consistent scheme.

2.5 Boundary treatment

Since this project concerns fluid sloshing in a tank, we will limit our discussion to no-slip or slip, stationary, rigid wall boundary. Treatment for other types of boundary conditions, such as periodic boundary, mass inlet and outlet, and moving object, can be found in Ref. [21, 45]. The four main types of methods to enforce boundary conditions in SPH are introduced as follows.

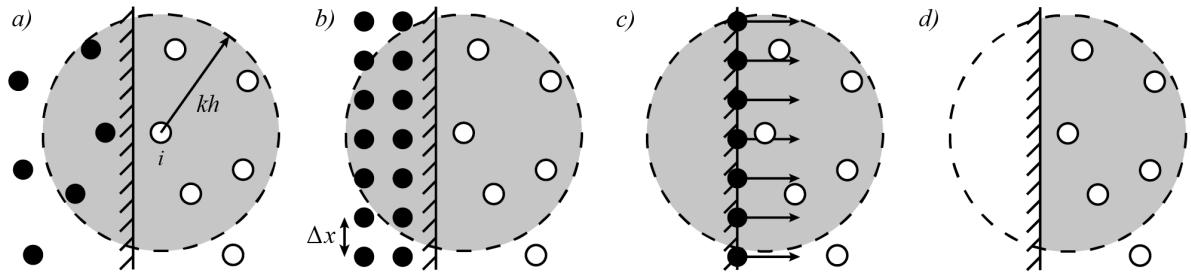


Figure 6: Different boundary treatment approaches: (a) mirror particles; (b) fixed particles; (c) repulsive particles; and (d) kernel renormalisation. Shaded area indicates the area of the kernel in normalisation.

a) Mirror/ghost particles (e.g. [17]): Fluid particles are mirrored across the wall, like the image in vortex method. To prevent penetration, the velocities of the mirrored particles normal

to the wall are equal and opposite to that of the corresponding fluid particles. The tangential velocities are flipped for no-slip boundary and remain the same for slip boundary. Because of the mirroring process, this technique is restricted to relatively simple domains with flat boundaries and convex geometries.

b) Fixed dummy particles (e.g. [1, 19, 50, 59]): Layers of dummy particles are placed outside the wall to provide kernel support for the fluid particles. These dummy particles are treated as fluid particles in computing time derivatives, just that their positions are not updated. Their density, velocity and pressure can be kept unchanged or be evaluated from local flow quantities. This type of boundary treatment is easy to implement, robust, and fairly accurate. Hence, it is the most commonly adopted method.

c) Repulsive force particles (e.g. [53, 56]): A layer of fixed particles are placed on the boundary, each of which exerts a radial repulsive force on the fluid particles. This method is easy to implement, suitable for all geometries, and computationally cheap. However, it is not very accurate and produces noisy fields. Also, it failed to simulate fluid in hydrostatic condition.

d) Semi-analytical renormalisation: This is a relatively new method, proposed by Ferrand *et al.* [29]. The method renormalises the solution by dividing the SPH summation by the volume of the kernel inside the domain, so that the condition in Eq.2 is satisfied.

$$[f_i] = \frac{1}{\Gamma} \sum_{j \in \mathcal{P}_i} f_j W_{ij} V_j, \quad \Gamma = \int_{\Omega^*} W_{ij} dV_j, \quad (25)$$

where Ω^* is the domain inside the boundaries (shaded region in Figure 6). This method is rigorous and accurate, and its solution is the least noisy among all boundary treatment methods. However, this method is hard to implement and is computationally expensive. Also, the volume integral requires analytical description of the boundaries, which can be cumbersome in 3D.

Table 1: Comparison between different boundary treatment methods

Method	Ease to implement	Accuracy	Computational efficiency	Suitability for complex domains
Mirror particles	✓✓	✓✓	✓✓	✓
Fixed particles	✓✓✓	✓✓	✓✓✓	✓✓
Repulsive particles	✓✓✓	✓	✓✓✓	✓✓✓
Renormalisation	✓	✓✓✓	✓	✓✓✓

Conclusion & decision: Having in mind (i) the limited computational resource and time for this project and (ii) boundary condition is not the main concern of this study, the *fixed dummy particle* approach is selected. It is simple to implement, adequately accurate, and computationally cheap. In particular, the method by Adami *et al.* [1] is used because it was reported to be the most accurate in its class [45]. The formulation is given as follows.

$$\mathbf{u}_l = 2\mathbf{u}_{w,l} - \frac{\sum_{j \in \mathcal{P}_l} \mathbf{u}_j W_{lj}}{\sum_{j \in \mathcal{P}_l} W_{lj}}, \quad (26)$$

$$p_l = \frac{\sum_{j \in \mathcal{P}_l} p_j W_{lj} + \mathbf{g} \cdot \sum_{j \in \mathcal{P}_l} \rho_j \mathbf{r}_{lj} W_{lj}}{\sum_{j \in \mathcal{P}_l} W_{lj}}. \quad (27)$$

The subscripts l and j refer to wall and fluid particles respectively, and \mathbf{u}_w is the prescribed wall velocity. Densities of the wall particles ρ_k are calculated using the pressure EoS (Eq.19).

2.6 Time integration scheme

There exists a class of time integration schemes that conserve total energy and linear and angular momentum of a Hamiltonian system, known as *symplectic*. Mathematically, the conservation property comes from the fact that the schemes preserve the volume of the state-space during mapping. As mentioned before, since conversation is extremely important for long-term accuracy and stability, symplectic schemes are widely implemented in SPH.

An n th-order symplectic scheme in Lagrangian reference frame can be generalised as [75]

$$\begin{aligned} \text{for } i = 0 \dots n-1 : \quad & \mathbf{x}^{i+1} = \mathbf{x}^i + c^i \mathbf{u}^i \delta t^i, \\ & \mathbf{u}^{i+1} = \mathbf{u}^i + d^i \mathbf{a}(\mathbf{x}^{i+1}) \delta t^i. \end{aligned} \quad (28)$$

Hence, an n th-order symplectic scheme requires evaluation of $\mathbf{a} = D\mathbf{u}/Dt$ for n times in each timestep $\Delta t = \sum_i \delta t^i$. However, the second-order scheme is a special case, where $c^0 = 0$. So, the acceleration from the previous timestep can be reused. This efficient symplectic scheme is the Störmer-Verlet scheme [83], also known as the kick-drift-kick scheme.

For all *fluid* particles,

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}^n + \frac{\Delta t}{2} \left(\frac{D\mathbf{u}}{Dt} \right)^n \quad (29a)$$

$$\mathbf{x}^{n+\frac{1}{2}} = \mathbf{x}^n + \frac{\Delta t}{2} \mathbf{u}^{n+\frac{1}{2}} \quad (29b)$$

$$\rho^{n+1} = \rho^n + \Delta t \left(\frac{D\rho}{Dt} \right)^{n+\frac{1}{2}} \quad (29c)$$

$$p^{n+1} = p(\rho^{n+1}) \quad (29d)$$

$$\mathbf{x}^{n+1} = \mathbf{x}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \mathbf{u}^{n+\frac{1}{2}} \quad (29e)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \left(\frac{D\mathbf{u}}{Dt} \right)^{n+1} \quad (29f)$$

For numerical stability, the timestep size Δt is limited by the following conditions [59]:

1. The CFL condition

$$\Delta t \leq 0.25C \left(\frac{h}{c_0 + \max(\|\mathbf{u}\|)} \right), \quad (30)$$

where $0 < C \leq 1$ is the CFL number.

2. The viscosity condition

$$\Delta t \leq 0.125 \left(\frac{h^2}{\nu} \right). \quad (31)$$

3. The body-force condition

$$\Delta t \leq 0.25 \sqrt{\frac{h}{\max(\|\mathbf{g}\|)}}. \quad (32)$$

3 Riemann solver based SPH

3.1 Modified equations of motion

There are two formulations to incorporate Godunov's scheme into SPH. The one from Vila [84] and Ben Moussa [4] computes fluxes directly at the midpoint of \mathbf{r}_{ij} . The other, first used by Inutsuka [41], evaluates the states at the midpoint of \mathbf{r}_{ij} . Cha [10, 11] showed that both formulations are equivalent. In this report, we will introduce the second formulation as it bears a similar form of the classical SPH, and hence, easier for comparison.

For each interacting particle pair, we form a 1D Riemann problem along \mathbf{r}_{ij} , as illustrated in Figure 7. The inter-particle averages are replaced by the solution of the Riemann problem at the midpoint of \mathbf{r}_{ij} , i.e. $(\mathbf{u}_i + \mathbf{u}_j)/2 \rightarrow \mathbf{u}^*$ and $(p_i + p_j)/2 \rightarrow p^*$. The equations of motion (Eq.22,24) can be rewritten as

$$\frac{D\rho_i}{Dt} = 2\rho_i \sum_j \frac{m_j}{\rho_j} (\mathbf{u}_i - \mathbf{u}^*) \cdot \nabla W_{ij}, \quad (33a)$$

$$\frac{D\mathbf{u}_i}{Dt} = -\frac{2}{\rho_i} \sum_j \frac{m_j}{\rho_j} p^* \nabla W_{ij}. \quad (33b)$$

3.2 Constructing stencils

3.2.1 Taylor series based extrapolation

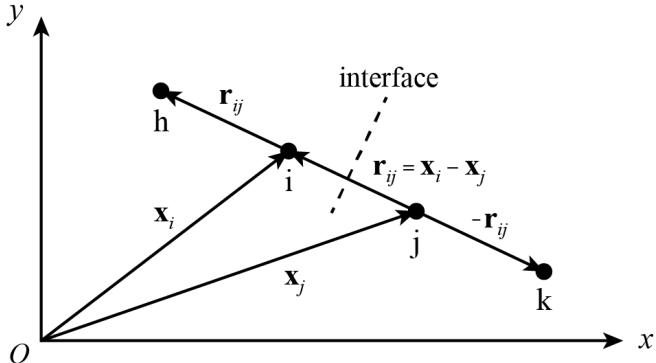


Figure 7: Schematics of the four-point stencil used in this work

In order to apply the high-order reconstruction schemes from Finite Volume (FV), we must construct an FV-like stencil. While a longer stencil allows us to employ higher-order reconstruction, it poses challenge in treatment near wall and free surface. In this study, a four-point stencil configuration is used because it is simple to implement and can achieve, in theory, up to fourth-order accuracy (can be fitted a third-order polynomial). In the literatures, there are three ways to obtain the values at h and k: (i) using the SPH interpolation operator (Eq.9) [41]; (ii) fitting a 2D polynomial in a Moving Least Square (MLS) approach [3]; and (iii) extrapolating using gradients (Eq.21) at i and j [42, 90]. We extend the third approach and propose an extrapolation method based on Taylor series expansion. This method is:

- **Computationally cheap and easy to implement** — it does not require additional range search as in the interpolation approach, and does not involve polynomial regression as in the MLS approach;
- **Flexible** — no special treatment is needed even if h and k fall outside of wall or free surface; and
- **Arbitrarily high-order** — its truncation error is $\sim \mathcal{O}(r_{ij}^{N+1})$ if up to the N th-degree derivatives are used.

The values at h and k are evaluated as

$$\begin{aligned} f_h &= f_i + \mathbf{r}_{ij}^\top \nabla f_i + \frac{1}{2} \mathbf{r}_{ij}^\top \mathbf{H}_i \mathbf{r}_{ij} + \mathcal{O}(r_{ij}^3), \\ f_k &= f_j - \mathbf{r}_{ij}^\top \nabla f_j + \frac{1}{2} \mathbf{r}_{ij}^\top \mathbf{H}_j \mathbf{r}_{ij} + \mathcal{O}(r_{ij}^3), \end{aligned} \quad (34)$$

where \mathbf{H} is the Hessian matrix

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial^2 y} \end{bmatrix}. \quad (35)$$

We also define the reduced Hessian as

$$\hat{\mathbf{H}} := \left[\frac{\partial^2 f}{\partial^2 x}, 2 \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial^2 y} \right]. \quad (36)$$

3.2.2 Advanced derivatives

To truly achieve the accuracy in Eq.34, we need to evaluate ∇f and \mathbf{H} as accurately as possible. This demand drives us to seek better derivative operators in SPH. Randles and Libersky [69], Bonet and Lok [6], and Vila [84] introduced a renormalisation tensor \mathbf{B} to the discretisation used in the continuity equation (Eq.22), making it first-order consistent, second-order accurate, and accurate near boundaries without specific treatment.

Table 2: Comparison between different discrete first derivative operators

Label	Expression ($\llbracket \nabla f_i \rrbracket =$)	Order of consistency	Order of accuracy*	Symmetric
F1	$\sum_j f_j \nabla W_{ij} V_j$	/	1	No
F2	$\sum_j (f_j - f_i) \nabla W_{ij} V_j$	0	1	No
F3	$\sum_j (f_j + f_i) \nabla W_{ij} V_j$	/	1	Yes
F4	$\sum_j (f_j - f_i) \mathbf{B}_i^{-1} \nabla W_{ij} V_j,$ $\mathbf{B}_i = - \sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j$	1	2	No

* For simplicity, the above table skipped some details on the order of accuracy of the schemes. Readers can refer to Ref. [27] for full descriptions.

On second derivatives, the most straightforward formulation is constructed using the second derivative of the kernel function (S1). However, since the second derivatives of most kernels are steep near $r_{ij} = 0$, it is sensitive to perturbation in the position of neighbouring particles. Also, the second derivative changes sign at the inflection point. This formulation is virtually

the same as using a kernel that is neither always positive nor monotonically decreasing, which could lead to instability [54]. There are a number of improvements: Korzilius [43] proposed the Corrective Smoothed Particle Method (CSPM), which renormalises the solution similar to that in F4; and Cleary's formulation [14] (S3), also known as the difference scheme, completely eliminates the use of kernel second derivative and successfully improved accuracy. Finally, the second derivative can also be found by differentiating the first derivative (S4).

Table 3: Different discrete second derivative operators

Label	Expression
S1	$\hat{\mathbf{H}}_i = \sum_j (f_j - f_i) \hat{\mathbf{H}}_{W,ij} V_j$
S2	$\hat{\mathbf{H}}_i = \sum_j (f_j - f_i - \mathbf{r}_{ij} \cdot \nabla f_i) \Theta_i^{-1} \hat{\mathbf{H}}_{W,ij} V_j,$ $\Theta_i = -\frac{1}{2} \sum_j \hat{\mathbf{H}}_{W,ij}^\top [r_{ij,x}^2, 2r_{ij,x}r_{ij,y}, r_{ij,y}^2] V_j$
S3	$\hat{\mathbf{H}}_i = \sum_j \left(\frac{f_j - f_i}{r_{ij}} \mathbf{e}_{ij} \cdot \nabla W_{ij} \right) (4\mathbf{e}_{ij} \otimes \mathbf{e}_{ij} - \mathbf{I}) V_j$
S4	$\mathbf{H}_i = \sum_j \mathbf{B}_i^{-1} [(\nabla f_j - \nabla f_i) \otimes \nabla W_{ij}] V_j$

Here, $\hat{\mathbf{H}}_{W,ij} = \left[\frac{\partial^2 W}{\partial^2 x}, 2 \frac{\partial^2 W}{\partial x \partial y}, \frac{\partial^2 W}{\partial^2 y} \right] = \frac{\partial^2 W}{\partial^2 r} [e_{ij,x}^2, 2e_{ij,x}e_{ij,y}, e_{ij,y}^2] + \frac{1}{r_{ij}} \frac{dW}{dr} [e_{ij,x}^2, -2e_{ij,x}e_{ij,y}, e_{ij,y}^2]$ is the reduced Hessian matrix of the kernel, and \mathbf{I} is the identity matrix.

Discussions

A convergence study is performed on some of the first and second derivative operators listed above. Attention has been paid to ensure that $\Delta x/h$ is sufficiently small. The scalar field

$$f(x, y) = \cos\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi y}{2}\right) - y. \quad (37)$$

is applied in the domain $\Omega \in [-1, 1] \times [-1, 1]$.

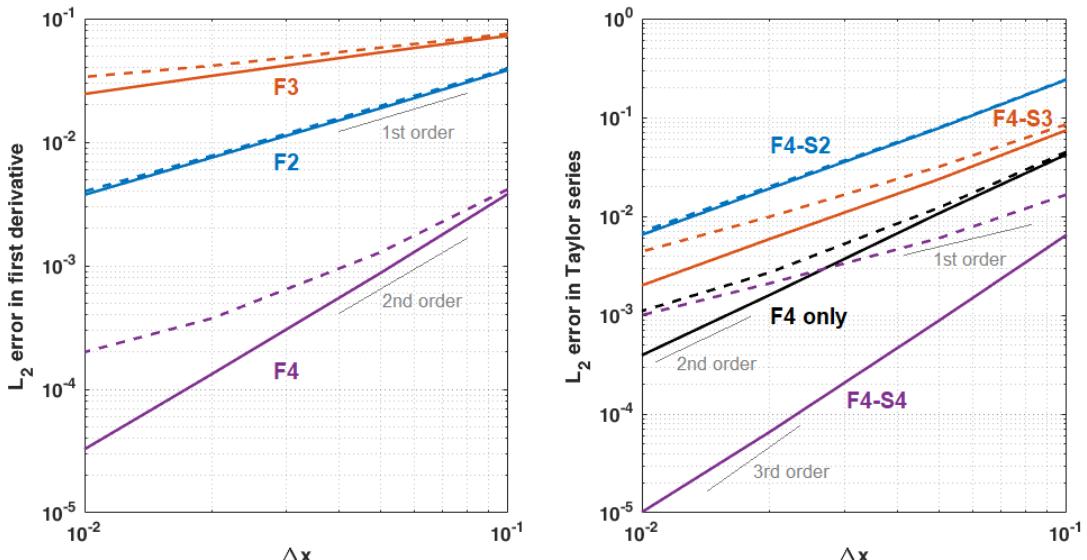


Figure 8: Convergence study of first and second derivative schemes with $\Delta x/h = 2/3$ and fifth-order Wendland kernel. (Solid line: regular points; Dashed line: 10% Δx perturbation applied.)

Figure 8 shows that F2 and F4 demonstrate the theoretical order of accuracy as presented in Table 2. While F3, suspected to be due to the low order of consistency, exhibits a reduction in accuracy in the field containing a linear term. F2 is resilient to particle disorder, but F4 is more accurate, despite having an about linear convergence with disordered particles. This is not surprising if we perform a simplified analysis on the central difference scheme in FD over a 1D non equi-spaced stencil.

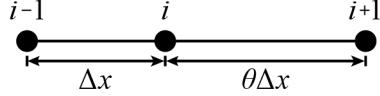


Figure 9: A general 1D three-point finite difference stencil

$$\begin{aligned}
 f_{i+1} &= f_i + f'_i \theta \Delta x + 0.5 f''_i \theta^2 \Delta x^2 + \mathcal{O}(\theta^3 \Delta x^3) \\
 -) \quad f_{i-1} &= f_i - f'_i \Delta x + 0.5 f''_i \Delta x^2 - \mathcal{O}(\Delta x^3) \\
 \hline
 f_{i+1} - f_{i-1} &= f'_i (1 + \theta) \Delta x - 0.5 f''_i (1 - \theta^2) \Delta x^2 + \mathcal{O}((1 + \theta^3) \Delta x^3) \\
 \frac{f_{i+1} - f_{i-1}}{(1 + \theta) \Delta x} &= f'_i + \underbrace{0.5 f''_i (\theta - 1) \Delta x}_{\text{truncation error}} + \mathcal{O}(\Delta x^2)
 \end{aligned} \tag{38}$$

$(\bullet)' := d \bullet / dx$. The same can be shown for second derivatives as well. For a smooth f , both first and second derivative schemes are second-order accurate if and only if the grid is uniform, i.e. $\theta = 1$. The convergence deteriorates to only first-order when perturbation is introduced to the position of the gridpoints. Although this analysis is only valid for FD, it is a useful analogy in understanding the behaviour of SPH, as they share the FD central differencing characterises. Rigorously, a similar analysis can be applied to SPH using Taylor series expansion, but this is out of the scope of this report.

On the overall extrapolation process, when particles are placed in a Cartesian grid, the F4-S4 scheme achieves a third-order accuracy and the scheme using only F4 first derivative reaches a second-order accuracy. The results confirm the error of the Taylor series expansion (Eq.34). However, as expected, the accuracy of both schemes degrade quickly to about first-order as soon as the particles are perturbed.

On the other hand, S2 and S3, suffering from excessive error near boundaries, perform even worse than linear extrapolation. Though, the study has provided evidence that the removal of kernel second derivative is beneficial to accuracy of SPH second derivative. In addition, it is observed that S3 display similar level of accuracy as S4 at the centre region, and its computational cost is about half of that of S2 and S4, as it does not include ∇f in the expression and hence can be performed in the same loop as the first derivative. This gives some incentives for future studies to search for better boundary treatment for S3.

In conclusion, although F2 is less sensitive to particle disorder, F4 is used to compute first derivative as it is more accurate. Scheme S4 offers various degrees of improvement in accuracy of the stencil, depending on particle distribution, but it comes with a considerable computational cost comparing to using F4 only. Therefore, our code provides a choice for users to switch on or off the second derivative, depending on one's computational resources. Note that all the results of the WENO scheme presented in this report will have the second derivative activated.

3.3 Reconstruction schemes

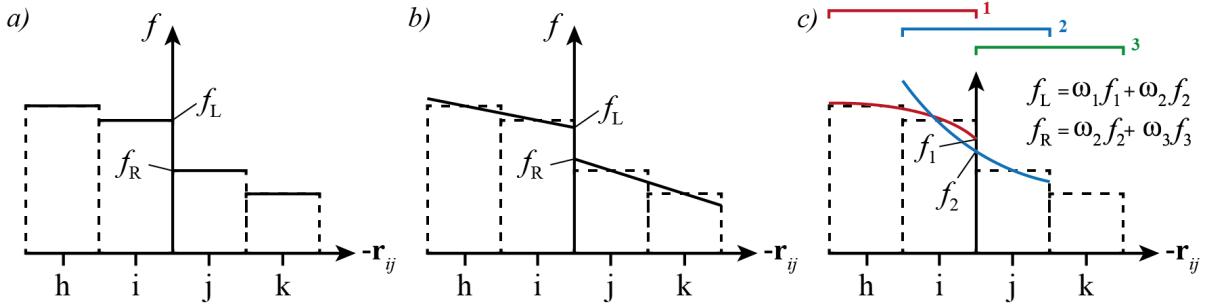


Figure 10: Different reconstruction schemes: (a) piecewise constant; (b) piecewise linear; and (c) piecewise parabolic (illustrated the reconstruction for left state only).

3.3.1 Baseline / Godunov's scheme

The baseline scheme is the same as the original Godunov's scheme, which assumes the flow quantities to be constant throughout the FV-like cells, hence is first-order accurate. We write the left and right states as \mathbf{W}_L and \mathbf{W}_R respectively.

$$\begin{aligned}\mathbf{W}_L &= [\rho_L, U_L, p_L] = [\rho_i, -\mathbf{u}_i \cdot \mathbf{e}_{ij}, p_i] = \mathbf{W}_i, \\ \mathbf{W}_R &= [\rho_R, U_R, p_R] = [\rho_j, -\mathbf{u}_j \cdot \mathbf{e}_{ij}, p_j] = \mathbf{W}_j.\end{aligned}\quad (39)$$

Note that the velocity \mathbf{u} is projected onto the direction of the Riemann problem.

3.3.2 Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL)

This scheme is originally proposed by van Leer [82]. It uses a linear extrapolation to evaluate quantities at the interface (f_L and f_R), and hence, is second-order accurate. However, for stability, the solution must be total variation diminishing (TVD). A physical interpretation of the TVD condition is that no new maximum or minimum is created. That is, the interfacial values must remain between the adjacent cell averaged values. Slope/flux limiters are introduced to enforce the TVD condition by returning the reconstructions that violate the TVD condition to first-order.

$$\begin{aligned}\mathbf{W}_L &= \mathbf{W}_i + \frac{1}{2} \Phi \left(\frac{\mathbf{W}_j - \mathbf{W}_i}{\mathbf{W}_i - \mathbf{W}_h} \right) \\ \mathbf{W}_R &= \mathbf{W}_j - \frac{1}{2} \Phi \left(\frac{\mathbf{W}_j - \mathbf{W}_i}{\mathbf{W}_k - \mathbf{W}_j} \right)\end{aligned}\quad (40)$$

where $\Phi(\mathbf{f}) := [\phi(f_1), \phi(f_2), \phi(f_3)]$ is the slope limiter. Connolly *et al.* [18] and Zhang [88] suggested that the superbee limiter by Roe [73] performs the best in SPH applications:

$$\phi(f) = \max[0, \min(2f, 1), \min(f, 2)]. \quad (41)$$

3.3.3 Weighted Essentially Non-oscillatory (WENO) scheme

As introduced in Section 1.2, the WENO scheme computes the left and right states as a weighted sum of all polynomials at the interface, where the non-linear weights are found using the smoothness indicators of the polynomials. A good WENO scheme should (i) have an accurate smoothness indicator; (ii) use compact stencils; and (iii) contain no tunable parameters/few parameters that are insensitive to types of problem. The WENO-Z by Borges *et al.* [7] and Castro *et al.* [9] satisfies the above conditions. It is third-order accurate. The three-point stencil, h-i-j, is divided into two two-point sub-stencils (1 and 2), as illustrated in Figure 10(c).

The interface value on the left is

$$f_L = \omega_1 f_1 + \omega_2 f_2, \quad (42)$$

where

$$f_1 = -\frac{1}{2}f_h + \frac{3}{2}f_i, \quad f_2 = \frac{1}{2}f_i + \frac{1}{2}f_j. \quad (43)$$

The smoothness indicators for parabola 1 and 2 are

$$IS_1 = (f_i - f_h)^2, \quad IS_2 = (f_j - f_i)^2, \quad (44)$$

respectively. And for $m = 1, 2$, the weights are given as

$$\omega_m = \frac{a_m}{a_1 + a_2}. \quad (45)$$

WENO-Z:

$$a_1 = \frac{1}{3} \left(1 + \frac{|IS_1 - IS_2|}{IS_1 + \epsilon} \right), \quad a_2 = \frac{2}{3} \left(1 + \frac{|IS_1 - IS_2|}{IS_2 + \epsilon} \right). \quad (46)$$

The right state, f_R , is obtained by reverting the stencils (using 2 and 3) and substituting the corresponding points: h \rightarrow k, i \rightarrow j, and j \rightarrow i.

3.4 Riemann solver

While the accuracy of a Godunov-type scheme is controlled by the order of its reconstruction, the Riemann solver handles the numerical dissipation. It takes in the left and right states of the Riemann problem and outputs the solution at the interface. Rafiee *et al.* [68] and Puri and Ramachandran [66] presented a number of approximate Riemann solvers in SPH context. For hydrodynamic and MHD applications, the Roe [73] and HLLC [80] solvers are commonly used in mesh-based CFD methods [65]. In this project, the Roe solver is selected because of its simplicity, low dissipation, and documented usage in WCSPH [40, 62, 90].

The Roe solver linearises the governing equations using the Jacobian matrix. The system is then diagonalised to three purely hyperbolic equations of the conservation variables to compute the solution in the desired region. Using $\gamma = 1$, the starred quantities are given as

$$\begin{aligned} U^* &= \frac{\rho_L c_{0,L} U_L + \rho_R c_{0,R} U_R + \alpha (p_L - p_R)}{\rho_L c_{0,L} + \rho_R c_{0,R}}, \\ p^* &= \frac{\rho_L c_{0,L} p_L + \rho_R c_{0,R} p_R + \beta \rho_L c_{0,L} \rho_R c_{0,R} (U_L - U_R)}{\rho_L c_{0,L} + \rho_R c_{0,R}}, \end{aligned} \quad (47)$$

$$\mathbf{u}^* = \frac{\mathbf{u}_i + \mathbf{u}_j}{2} + \left(\frac{U_L + U_R}{2} - U^* \right) \mathbf{e}_{ij}. \quad (48)$$

Here, α is a novel diffusion limiter introduce to avoid surface rising (see Section 5.1), and β is the dissipation limiter introduced by Zhang [90] to achieve low numerical dissipation:

$$\alpha = \begin{cases} 0, & |p_L - p_R| \leq \|\mathbf{g}\|kh \\ 1, & |p_L - p_R| > \|\mathbf{g}\|kh \end{cases} \quad (49)$$

$$\beta = \min[1, 3 \max((U_L - U_R)/\bar{c}, 0)], \quad (50)$$

with $\bar{c} = (\rho_L c_{0,L} + \rho_R c_{0,R})/(\rho_L + \rho_R)$. Zhang's limiter removes dissipation when the fluid is under the action of a rarefaction wave, i.e. when $U_L < U_R$.

If $c_{0,L} = c_{0,R}$ and $\rho_L = \rho_R$, the implicit dissipation can be clearly observed, as

$$\begin{aligned} \mathbf{u}^* &= \frac{\mathbf{u}_i + \mathbf{u}_j}{2} - \frac{1}{2} \alpha \left(\frac{p_L - p_R}{\rho c_0} \right) \mathbf{e}_{ij}, \\ p^* &= \frac{p_L + p_R}{2} + \frac{1}{2} \beta \rho c_0 (U_L - U_R). \end{aligned} \quad (51)$$

3.5 Modified boundary treatment

Since wall particles are outside of the fluid domain, derivatives of the flow quantities at wall particles are ill-defined. In practice, there also lacks complete kernel support for calculating derivatives at wall particles—we cannot have wall particles of wall particles. Luckily, Dubois [22] proposed the so-called ‘partial Riemann problem’, that is, a Riemann problem in which one side is subjected to boundary conditions. He showed that the partial Riemann problem can be reformulated into a typical Riemann problem. This technique is applied to SPH in Ref. [88–91].

To form the partial Riemann problem, the stencil must be placed normal to the wall. The wall normal vector is calculated as

$$\mathbf{n}_{w,l} = -\frac{\sum_{m \in \mathcal{P}_l} \nabla W_{lm}}{\|\sum_{m \in \mathcal{P}_l} \nabla W_{lm}\|}. \quad (52)$$

Note that the summation is over wall particles only, so, if the wall does not deform, the wall normal vector only needs to be evaluated once, at the beginning of the simulation.

The boundary conditions for our wall is no-slip and impermeability. The left (fluid) states are calculated as $[\rho_L, U_L, p_L] = [\rho_i, -\mathbf{u}_i \cdot \mathbf{n}_{w,l}, p_i]$, and the right states are

$$U_R = -2(\mathbf{u}_{w,l} \cdot \mathbf{n}_{w,l}) - U_L, \quad (53)$$

$$p_R = p_L + \rho_L \mathbf{g} \cdot \mathbf{r}_{lj}, \quad (54)$$

and ρ_R is calculated from p_R using the EoS (Eq.19).

Although not being an emphasise of this project, the method can also accommodate moving walls, for example, in the lid-driven cavity simulation presented in Appendix A.

4 Implementation

The Riemann solver based SPH method described in the previous sections is implemented in MATLAB. The Parallel Computing Toolbox in MATLAB is applied to allow some level of parallelisation, although the code has not been optimised for parallel performance. The source code, documentation, and the numerical test cases can be found in the GitHub repository: <https://github.com/ensonun/eSPH>.

5 Numerical tests

5.1 Hydrostatic case

The hydrostatic case is perhaps the simplest test case for CFD codes. Although it may seem trivial, obtaining a stable hydrostatic solution is a challenging task for SPH. Below, we will present and discuss a hydrostatic case with a rectangular tank. For comparison, we will display our results alongside the δ -SPH code developed by Jan Robert Eichstädt [24].

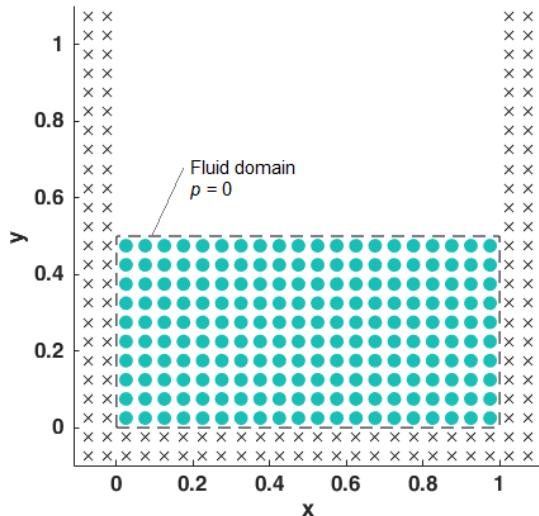


Figure 11: Setup of the hydrostatic case

Table 4: Parameters in the hydrostatic case

Parameter	Value
Kernel	Wendland
Δx	0.05
$h/\Delta x$	1
ρ_0	1
m	$\rho_0 \Delta x^2$
v_{max}	\sqrt{gH}
γ	7
\mathbf{g}	$[0, -1]^\top$
CFL number	1

Monaghan and Kajtar [56] and Adami *et al.* [1] suggested that a step change in gravitational force will lead to spurious pressure oscillation. Hence, a damping function f_d is introduced to warm the system up smoothly. In this test case, we take $t_d = 1$.

$$f_d(t) = \begin{cases} 0.5 \sin\left(\frac{t}{t_d} - 0.5\right) - 0.5 & t < t_d, \\ 1 & t \geq t_d. \end{cases} \quad (55)$$

The total energy E is comprised of kinetic energy KE , potential (or external load) energy PE , and internal energy IE ,

$$KE = \sum_j \frac{1}{2} m_j \|\mathbf{v}_j\|, \quad (56)$$

$$PE = \sum_j m_j \mathbf{g} \cdot \mathbf{x}_j, \quad (57)$$

$$IE = \int_{\rho_0}^{\rho} \frac{mp}{\rho^2} d\rho = \sum_j \frac{m_j c_0^2}{\gamma - 1} \left(\left(\frac{\rho_j}{\rho_0} \right)^{\gamma-1} + \frac{\rho_0}{\rho_j} (\gamma - 1) - \gamma \right), \quad (58)$$

where the summation is over all fluid particles.

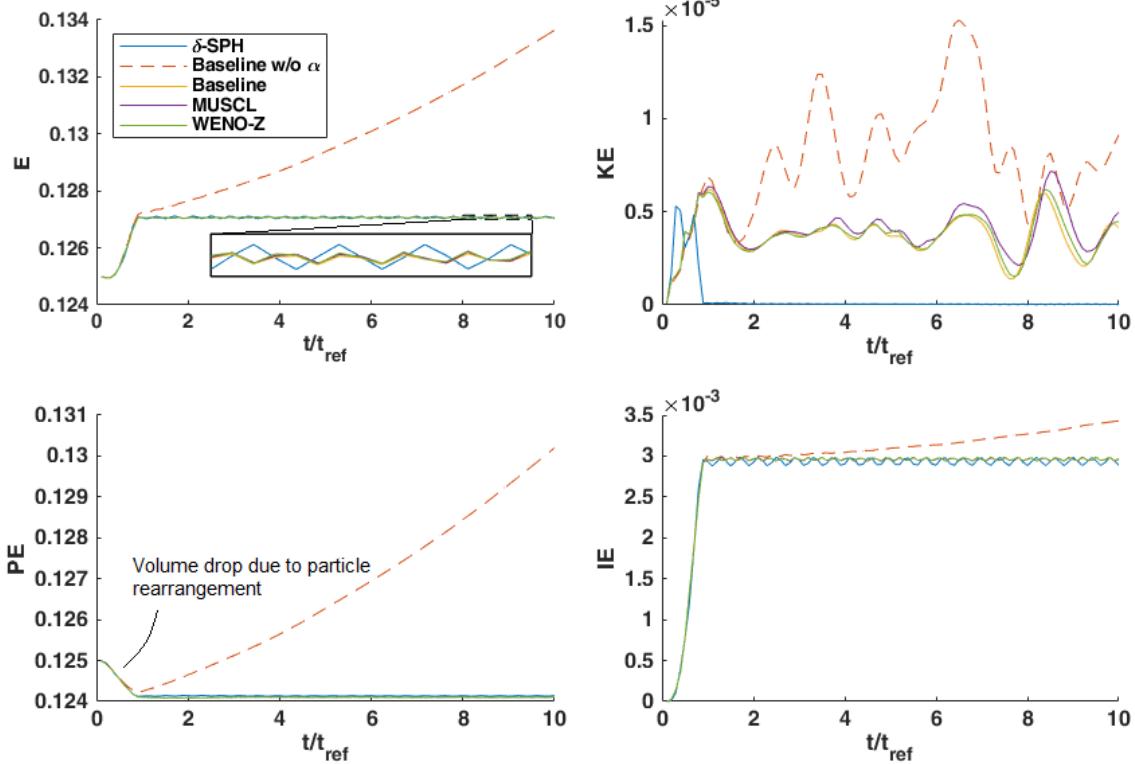


Figure 12: Energy evolution of δ -SPH and different reconstruction schemes. (Notes: δ -SPH uses $\alpha = 0.24$, $\delta = 0.02$ as suggested by [24]; Second derivative is switched on for stencil extrapolation in WENO-Z; and $t_{ref} = \sqrt{g/H}$ is the reference time scale.)

5.1.1 Artificial diffusion limiter

The Riemann solver SPH without the diffusion limiter exhibits an unphysical increase in total energy, mainly contributed by the rising of fluid surface. The pressure distribution in Figure 13 further reveals the error near free surface. This phenomenon is caused by the artificial diffusion term in the Roe Riemann solver, which is proportional to the pressure difference $p_L - p_R$. As illustrated by Figure 15, a particle at the centre and on the bottom of the tank sees a zero net artificial diffusion as the left-right pressure difference is cancelled out. However, a near-surface particle always sees a negative net artificial diffusion because of the incomplete kernel support. As a result, the particle’s $D\rho/Dt$ is overestimated, and hence, the pressure. The higher pressure of the layer(s) of particles right below the surface ‘pushes’ the surface particles upwards,

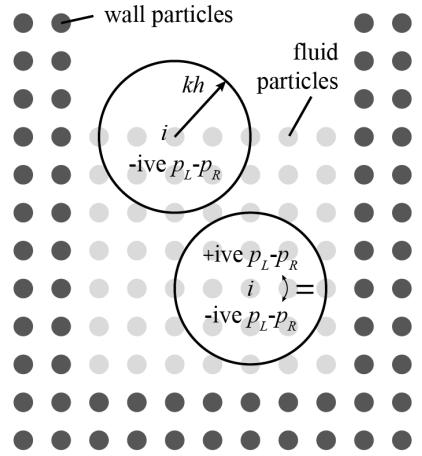


Figure 15: Near-surface particles see a net negative $p_L - p_R$

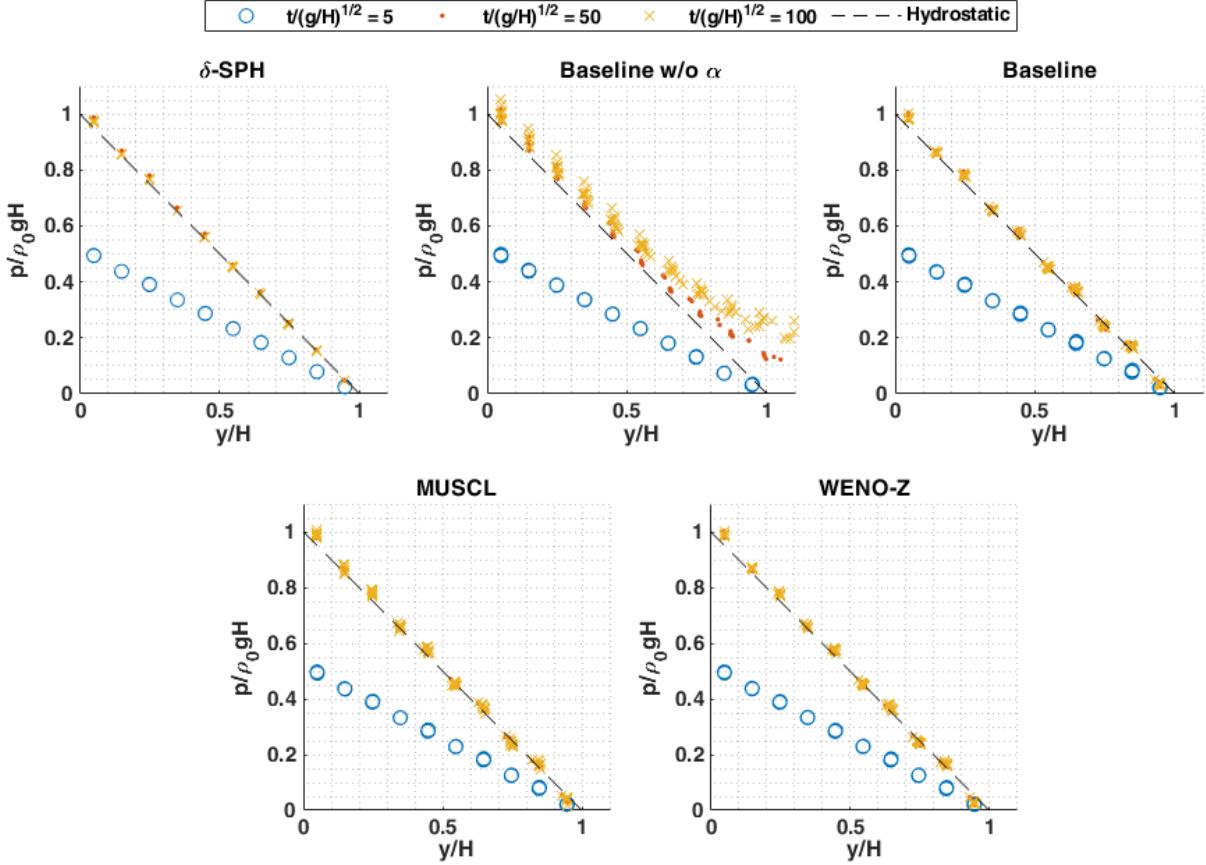


Figure 13: Normalised pressure vs normalised particle height (H is fluid initial height)

producing the surface rising. As long as the free surface is present, the problem will continue to grow. Therefore, the diffusion limiter (α) we proposed switches off the artificial diffusion when $p_L - p_R$ is less than the hydrostatic pressure difference. However, this fix has two side effects.

The first being the undamped (or slowly damped) kinetic energy as comparing to the δ -SPH. The effect is also observed as a slightly noisier pressure distribution in Figure 13. Because the simulation is relatively short (long-term simulation usually refers to t/t_{ref} of over 200), we cannot confirm whether or not the kinetic energy will vanish when the particles finished rearranging into a more stable triangular distribution. If the kinetic energy remains at a constant level, it could lead to some instabilities that require other countermeasures.

The second consequence of the diffusion limiter is that the spurious modes, as shown in Figure 14, are not sufficiently suppressed. The chessboard mode in pressure and the velocity current are two well-observed numerical artefacts in SPH [5, 28]. The chessboard mode can be understood as a spatial periodic mode with wavelength of $2\Delta x$. We again employ the FD central-difference derivative analogy,

$$\left(\frac{df}{dx} \right)_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x}. \quad (59)$$

The mode with exactly $2\Delta x$ wavelength yields a zero derivative. Thus, it will not be ‘seen’ by the system, but will remain in the solution. As observed in Figure 14, the dark-and-light bands pattern confirms that the wavelength of the mode is $2\Delta x$. Furthermore, the velocity current is

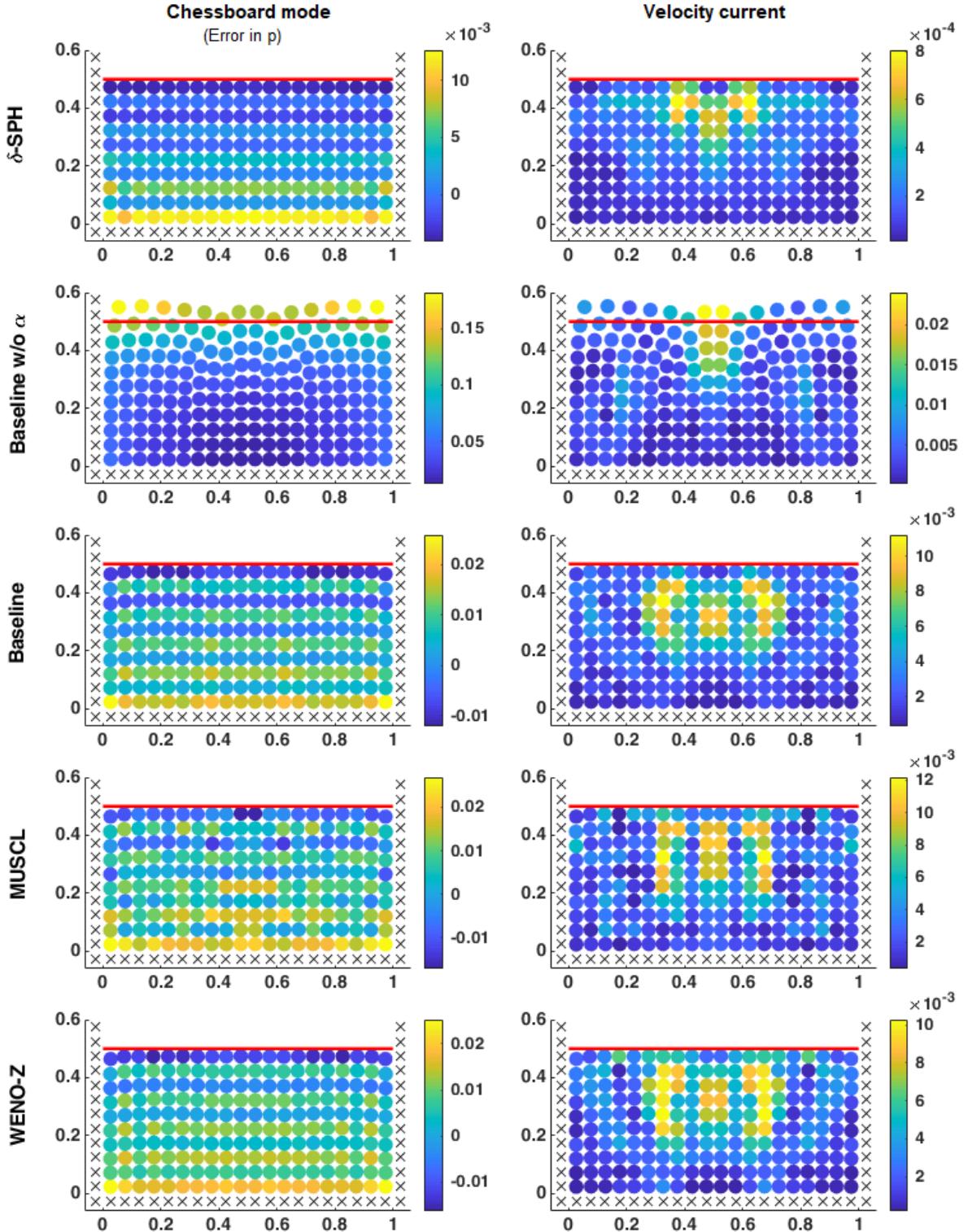


Figure 14: Error field of different schemes at $t/t_{ref} = 100$

due to particle rearrangement. As the initial Cartesian grid is an unstable layout for particles, together with the decrease in total volume due to pressure rise, the particles tend to move and rearrange themselves into a more stable triangular grid. This is a stable process on its own, but without dissipation, the velocity could trigger other instabilities.

Despite the issues discussed above, the Riemann solver SPH with all three reconstruction schemes and the diffusion limiter successfully produced solutions with stable total energy, low kinetic energy, and accurate pressure in this challenging hydrostatic problem.

5.2 Dam-breaking

The dam breaking problem is one of the most commonly used test cases in SPH. It involves violent impact, complex free surface and wave breaking, and therefore is ideal for assessing the stability, dissipation level, and boundary treatment of the SPH code.

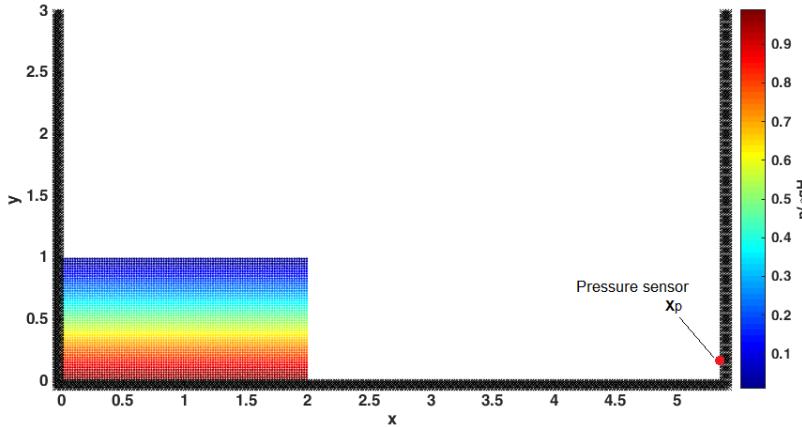


Table 5: Parameters in the dam-breaking case (Unspecified parameters remain the same as the previous case.)

Parameter	Value
Δx	0.02
$h/\Delta x$	1.5
v_{max}	$2\sqrt{gH}$

Figure 16: Setup of the dam-breaking case at $t = 0$

5.2.1 Comparing to analytical and numerical works

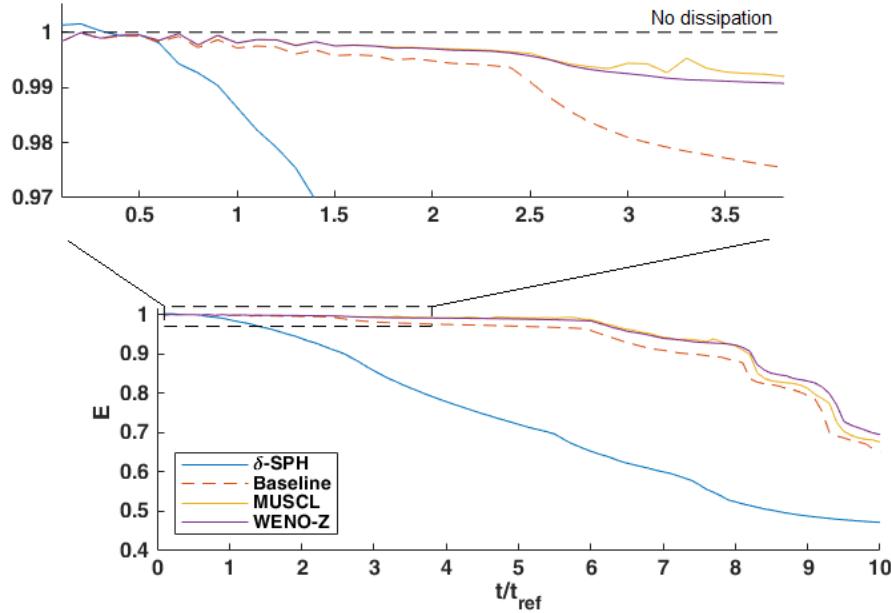


Figure 17: Energy evolution of δ -SPH ($\alpha = 0.1$, $\delta = 0.02$) and Riemann solver SPH with different reconstruction schemes in the dam-breaking case

Our simulations are inviscid, which means total energy should be conserved. Any loss in en-

ergy must therefore be caused by numerical dissipation of the schemes. As shown in Figure 17, all Riemann solver SPH experience significantly less dissipation than the reference δ -SPH throughout the simulation time. This result matches our objective. Moreover, since dissipation is proportional to the difference in left and right states of the Riemann problem, and the TVD condition restricts that the reconstructed states must remain inside the range of the piecewise constant states, higher-order reconstruction schemes will only reduce the left and right state difference, and hence, yield lower levels of dissipation. This is confirmed in the energy plot, as the MUSCL and WENO-Z conserve energy better than the baseline scheme. Note that the dissipation is done implicitly, completely without user input and information about the problem.

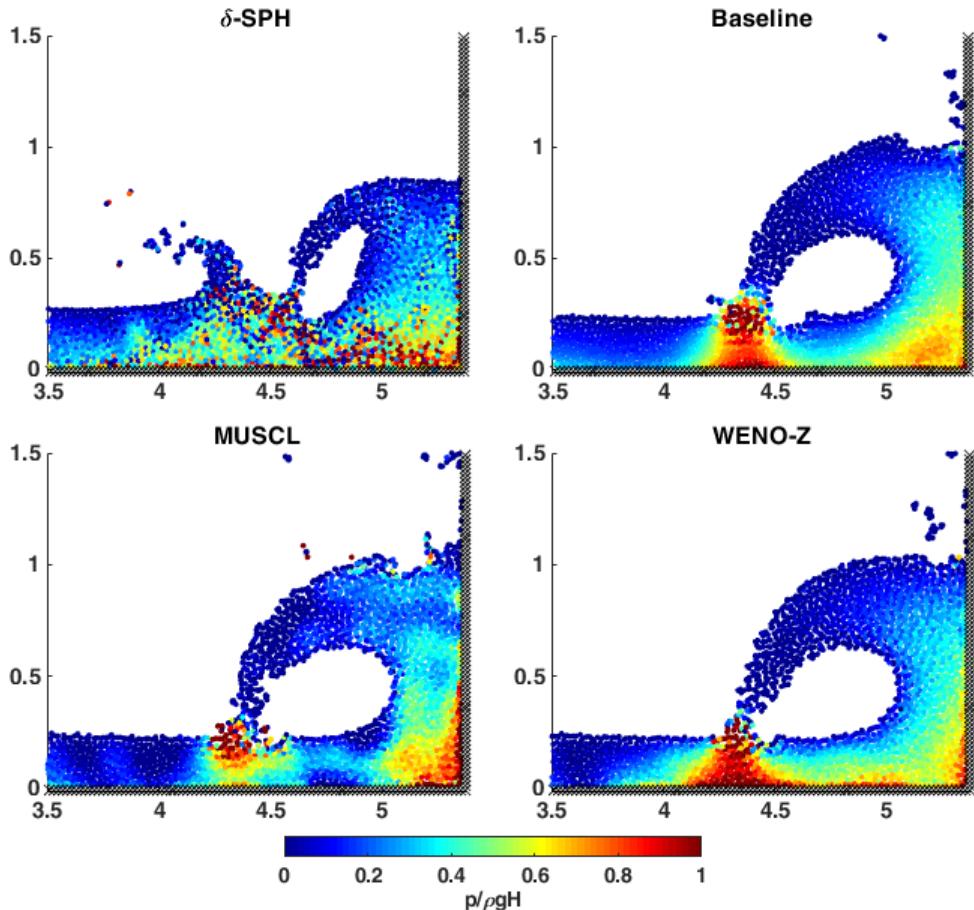


Figure 18: Pressure field of different schemes at $t/t_{ref} = 6.1$

However, a low dissipation does not mean anything on its own, we must also examine the solution to see whether the dissipation is sufficient for stabilising the solution, and whether the simulation correctly recreates the flow structures. Figure 18 reassures us that the simulated pressure field is smooth and stable. Some noise is observed near the top right corner of the wave in MUSCL scheme's solution. That is due to a few particles being shot up to the air now drop back to the fluid. It is not instability. Figure 19 compares the snapshots of our Riemann solver SPH against the highly-cited SPH simulation by Adami *et al.* [1]. Our results demonstrate good agreement in free surface shape and pressure field with the reference simulation at all time instances. No leakage or instability is observed throughout the simulation. Looking at the wavefront position (found as the position of the rightmost particle) in Figure 20, the solutions of the Riemann

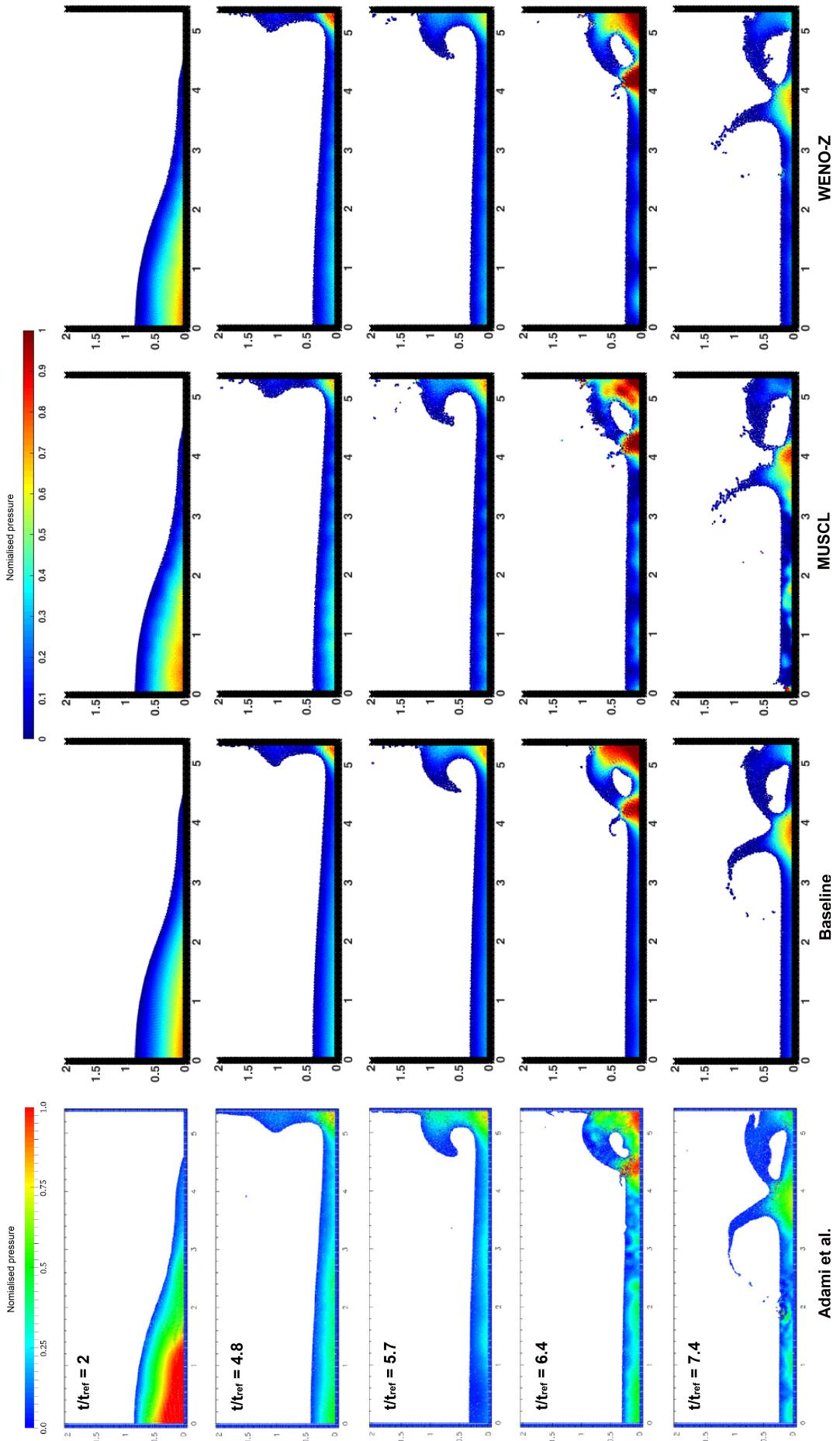


Figure 19: Snapshots of the normalised pressure field in dam-breaking case (Leftmost: Adami et al. [1])

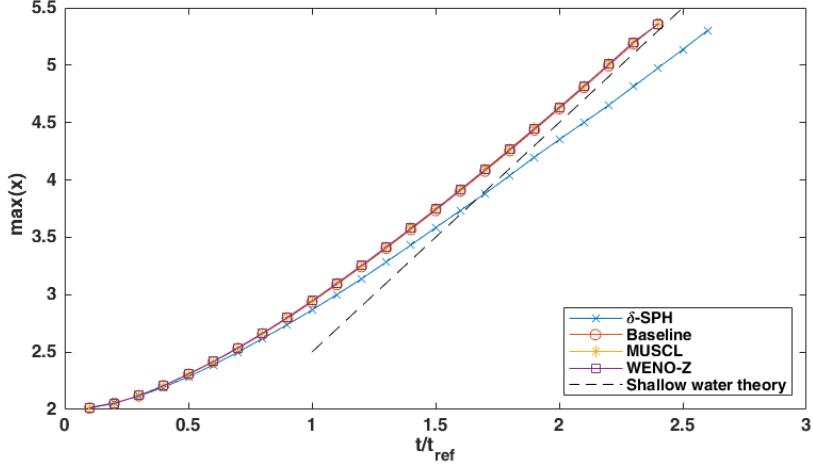


Figure 20: Wavefront location of different schemes against analytical result [71]

solver SPH matches closely with the analytical solution from shallow water theory [71]. While the δ -SPH result, because of excessive dissipation of energy, has a much lower wavefront speed.

5.2.2 Comparing to experimental data

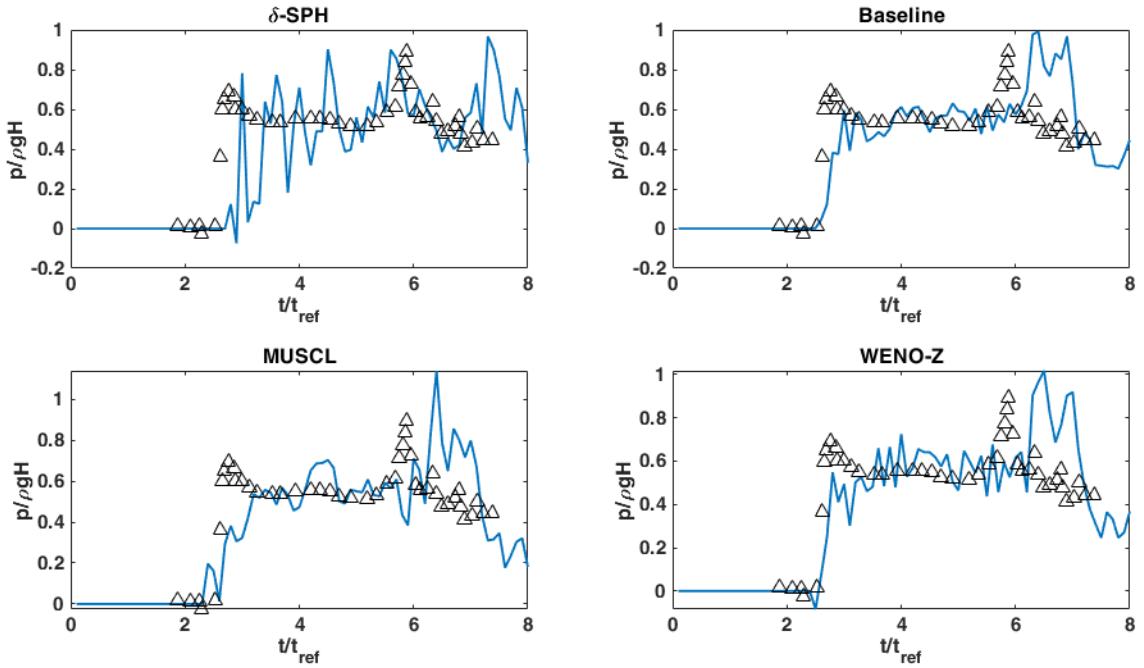


Figure 21: Pressure evolution at $\mathbf{x}_p = [5.36, 0.2]^\top$ of different schemes in dam-breaking case comparing to the experimental data from Buchner [8] (black triangles)

A pressure sensor is employed to record the pressure variation over time. The results are plotted in Figure 21. The pressure sensor measures the local pressure by SPH interpolation, i.e.

$$p_p = 2 \sum_j \frac{m_j}{\rho_j} p_j W_{pj}, \quad (60)$$

where $\mathbf{x}_p = [5.36, 0.2]^\top$ is the location of the sensor, and all other symbols carry their usual meaning. The summation is multiplied by two because only half of the kernel is in the domain.

Three observations can be made. Firstly, the large-amplitude pressure fluctuation in δ -SPH is well damped by the dissipation in the Riemann solver. Secondly, all Riemann solver SPH accurately capture the impact time ($t/t_{ref} \approx 2.4$), rise-time in pressure, the pressure level after impact, and the later pressure surge. However, the pressure surge predicted is about 0.4 unit time delayed. This phenomenon is shared by many other 2D SPH simulations (e.g. [1, 24, 34, 90]) of this case, hence, the possibility of it being caused by numerical error is eliminated. Other factors, such as 3D effects, viscosity, and a different initial condition, may be responsible for this discrepancy. Thirdly, among all three Riemann solver SPH, the MUSCL and WENO-Z schemes are slightly noisier than the baseline scheme, although the level of noise is acceptable. It is suspected to be a tradeoff for lower numerical dissipation.

5.3 Oscillating droplet in a conservative force field

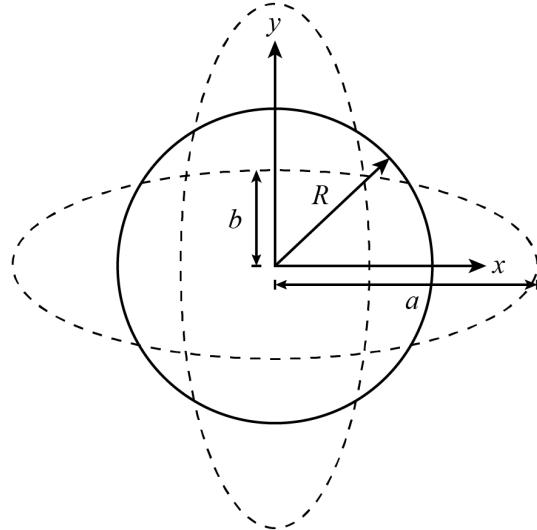


Figure 22: Setup of the oscillating droplet case. R is the initial radius of the droplet, $a(t)$ and $b(t)$ are the semi-major and semi-minor axis of the evolving ellipse, respectively. For SPH simulations, $v_{max} = 1.5A_0$.

Initially proposed by Monaghan and Rafiee [58], this problem is often used as a benchmark for SPH schemes thanks to the existence of the analytical solution. In the force field of

$$\mathbf{g}(x, y) = -A_0^2 [x, y]^\top, \quad (61)$$

the initially circular droplet oscillates between the two ellipses as illustrated in Figure 22. The initial velocity and pressure of the droplet are

$$\mathbf{u}(x, y, 0) = A_0 [x, -y]^\top, \quad p(x, y, 0) = \frac{1}{2}\rho_0 A_0^2 (1 - x^2 - y^2). \quad (62)$$

The dynamics of the system is governed by the non-linear ODE of $A(t)$ and $a(t)$:

$$\frac{d}{dt} \begin{bmatrix} A \\ a \end{bmatrix} = \begin{bmatrix} (A^2 - A_0^2) \left(\frac{R^4 - a^4}{R^4 + a^4} \right) \\ Aa \end{bmatrix}. \quad (63)$$

The equation can be integrated numerically, provided the initial conditions $A(0) = A_0 = 1.5$ and $a(0) = R = 1$.

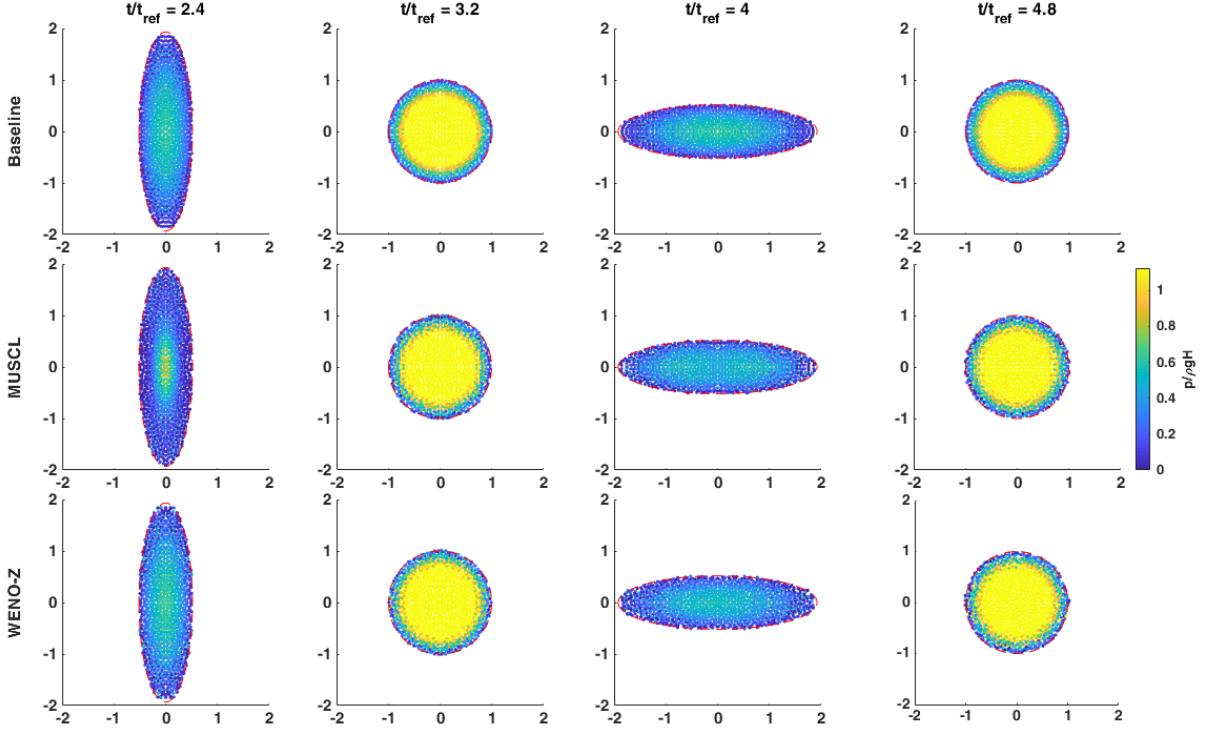


Figure 23: Snapshots of the oscillating droplet with $\Delta x = 0.05$. All cases show good agreement with the analytical shape (red dashed line), and the pressure fields are smooth.

5.3.1 Convergence study

Choosing $\Delta x/h$: As mentioned in Section 2.4, for assessing the convergence of SPH schemes, one must ensure that $\Delta x/h$ is sufficiently small for the convergence curve to enter the asymptotic region. First, we choose the points of Δx for the convergence study to be 0.1, 0.05, 0.03, and 0.02. This is based on the computational power available, and in an attempt to spread the points in the interval evenly. Then, simulations are performed using the finest Δx , WENO-Z, and with three different $\Delta x/h$ using the Wendland kernel. The error is plotted against $\Delta x/h$, and $\Delta x/h = 0.67$ is deemed suitable as the error stops decreasing. It has to be admitted that this study is relatively loose, as too few points are used, limited by time and computational resources. But looking at other WENO-SPH schemes, Avesani [3] used a $\Delta x/h$ of 0.5, and Zhang [91] used 0.77, which are close to our value and can justify our choice.

Using the geometrical identity for ellipses, $a(t)b(t) = R^2$, the L_q error is defined as

$$L_q = \frac{1}{N} \left(\sum_{n=1}^N |a_n b_n - R^2|^q \right)^{1/q}, \quad (64)$$

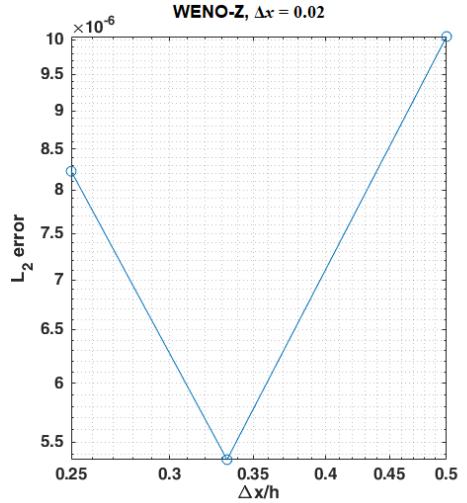


Figure 24: Searching for a sufficiently small $\Delta x/h$

where q can be any positive integer, and n are the timesteps used, in our case, the $t/t_{ref} \in [3.2, 4.9]$.

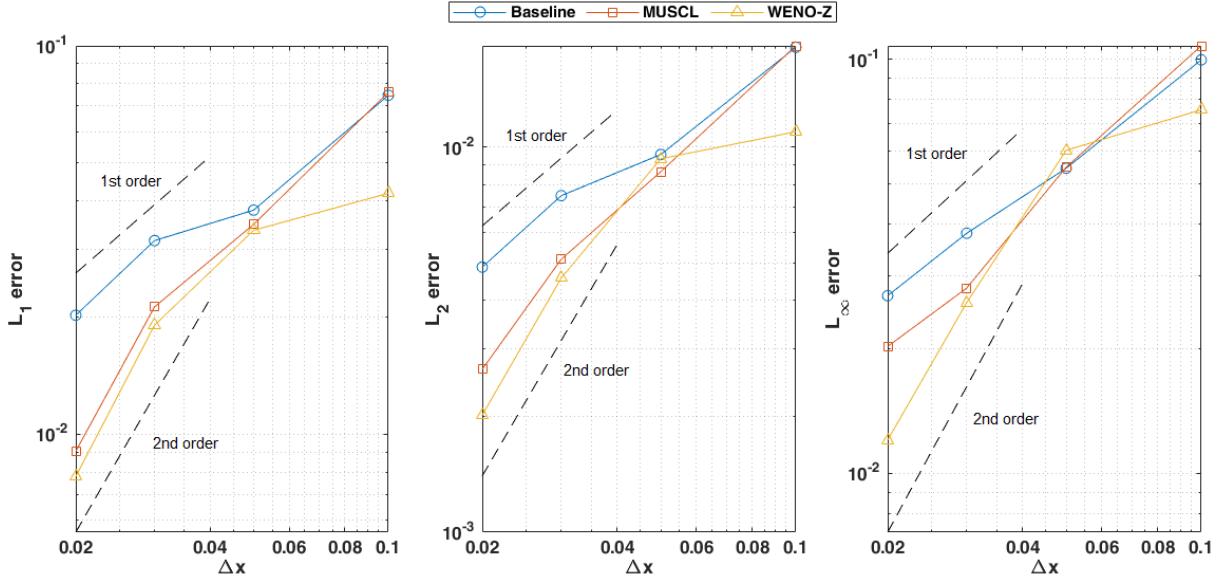


Figure 25: Convergence of different schemes in oscillating droplet case ($h/\Delta x = 1.5$)

The errors are plotted in Figure 25. An immediate observation is that the lines are not straight, but rather curves with steeper gradients as the Δx reduces. This is because when particles are too coarse, the error will not increase indefinitely. Instead, it will tend to a constant level, so we see a less steep slope of convergence. If we only consider the finest two points, which of course is a straight line, it reveals a different picture that is consistent with the theory.

Table 6: Errors and order of convergence in the finest range.

	Δx	L_1	L_2	L_∞	$\mathcal{O}(L_1)$	$\mathcal{O}(L_2)$	$\mathcal{O}(L_\infty)$
Baseline	0.03	3.15E-2	7.49E-3	3.89E-2			
	0.02	2.03E-2	4.88E-3	2.69E-2	1.09	1.05	0.77
MUSCL	0.03	2.13E-2	5.12E-3	2.80E-2			
	0.02	9.05E-3	2.65E-3	2.02E-2	2.11	1.62	1.10
WENO-Z	0.03	1.91E-2	4.58E-3	2.58E-2			
	0.02	7.79E-3	2.02E-3	1.20E-2	2.21	2.02	1.76

From Table 6, higher-order reconstruction schemes always give more accurate results than the lower ones. The baseline scheme is generally first-order accurate, which matches the fact that it reconstructs the solution using zeroth-order polynomials (piecewise constant). The MUSCL scheme is second-order accurate. Although the L_∞ error, i.e. maximum error, converges slower, consider the fact that this is the case for all schemes, we still see an about 50% improvement in order of accuracy comparing to the baseline scheme.

The WENO scheme, despite having the highest accuracy and the highest rate of convergence, fails to meet the third-order accuracy promised by the reconstruction. This is because in practical simulations, when particles are highly disordered, the stencil construction process becomes

inaccurate for the reasons discussed in Section 3.2.2, which limits the performance of the WENO scheme. Possible ways for improvement will be discussed in Section 6.1.

Limitations of the convergence study: Firstly, the particles are clearly not fine enough to reach the asymptotic convergence. The results using only two points may be not convincing enough, even though it agrees well with the theory.

Secondly, the oscillating droplet case is dynamic. There are both spatial and temporal errors in the solution. The way we calculated the error is a combination of the two. Consider a composite error $\delta\tilde{x} = x_{\text{numerical}} - x_{\text{exact}}$. Under the perturbation of some spatial error δx and some temporal error δt ,

$$\delta\tilde{x} \approx \delta x - \frac{\partial x}{\partial t}\delta t. \quad (65)$$

The error could be underestimated if δx and $\frac{\partial x}{\partial t}\delta t$ have the same sign. A better approach is to separate the spatial and temporal errors. For example, by a Hilbert transform, we can track the phase of the numerical result and relate it to the point with the same phase in the analytical result. It would be interesting to see how both errors converge with different schemes.

Thirdly, the simulation is run for a relatively short time. A longer simulation will reduce the contribution of random error, thus revealing the true performance of the numerical schemes.

Fourth and finally, the convergence is only confirmed for this particular case and with the particular set of A_0 and R . Studies on more cases should be performed to confirm the results.

5.3.2 Computational efficiency

The computations are performed on PCs at Imperial College London, using 4 cores of Intel Core i7-6700 3.40GHz CPU and 8GB of memory. The runtime is measured using MATLAB built-in timer function. The environment is not quiet enough for exact measurements, but the results can be indicative.

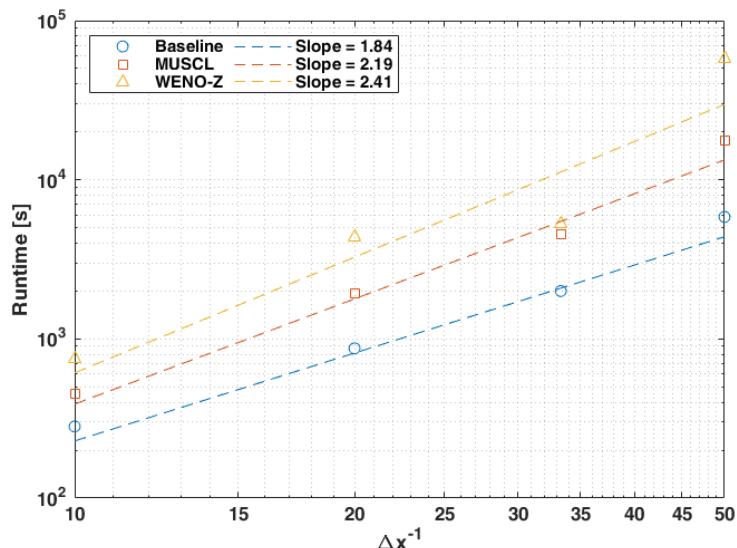


Figure 26: Runtime of different schemes against particle resolution

Computational efficiency is defined as $1/(\text{Runtime} \times \text{Error})$, as we want to minimise the two. The computational efficiency of the baseline, MUSCL, and WENO-Z schemes are -1.07 to -0.75, -1.09 to -0.08, and -0.65 to -0.2 orders of Δx , respectively. The higher (less negative) the order the better. In the best scenario, MUSCL is the most efficient scheme. The efficiency of the WENO-Z scheme with second-order derivatives is always better than the baseline scheme. If second-order derivatives are disabled, the WENO scheme will have the same order of computational cost as the MUSCL scheme. A convergence test can be performed to see if it can achieve better efficiency than the MUSCL scheme or not.

In summary, the results in this section show that our high-order reconstruction schemes improved both accuracy and efficiency in the Riemann solver SPH framework.

5.4 Linear sloshing in a rectangular tank

Echoing the title of this report, the primary motivation for using SPH is its capability in simulating free surface. This section will be a short assessment of the ability of our code in capturing free surface. To do so, we choose the linear sloshing case as it is one of the few test cases that involve free surface and can be solved analytically (at some limit of small perturbation).

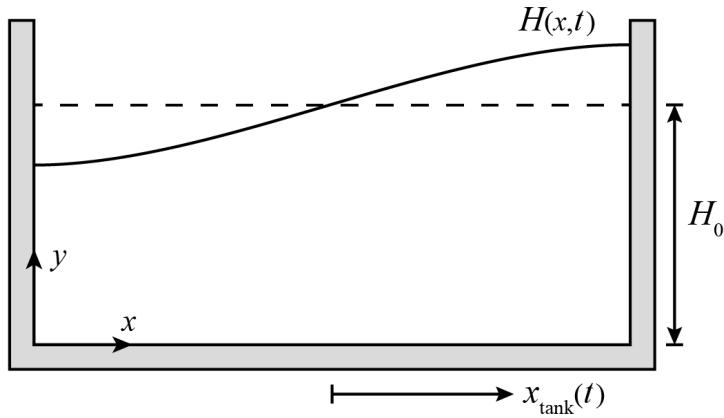


Figure 27: Setup of the linear sloshing case

The initially hydrostatic tank is moved in the x -direction, with the displacement $x_{\text{tank}}(t) = A \sin(2\pi f_e t)$, where A is the amplitude of oscillation (can be non-dimensionalised) and f_e is the excitation frequency. Changing the frame of reference to that moving with the tank, now the tank sees no displacement, but the fluid inside it experiences a horizontal acceleration equal to \ddot{x}_{tank} . This acceleration can be added to the gravity, as

$$\mathbf{g}(t) = \begin{bmatrix} -(2\pi f_e)^2 A \sin(2\pi f_e t) \\ -g \end{bmatrix}. \quad (66)$$

The analytical solution for fluid surface height (measured from the bottom of the tank) is given by Faltinsen [26], as

$$H(x, t) = H_0 + \sum_{M=0}^{\infty} \xi_M(t) \cos\left(\frac{M\pi x}{L}\right), \quad H(x, 0) = H_0. \quad (67)$$

For even M , $\xi_M(t) = 0$. For odd number M , $\xi_M(t)$ is obtained by solving the ODE

$$\ddot{\xi}_M + (2\pi f_M)^2 \xi_M = \frac{4A(2\pi f_e)^2}{M\pi} \sin(2\pi f_e t) \tanh\left(\frac{M\pi H_0}{L}\right). \quad (68)$$

This is solved numerically, for up to $M = 25$. The first natural frequency of the system is

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{\pi g}{L} \tanh\left(\frac{\pi H_0}{L}\right)}. \quad (69)$$

In the following simulations, we set $H_0 = 0.4$, $L = 1$, $g = 1$, $A = 0.25$, and $f_e = 0.3f_0$. This set of parameters are chosen such that the sloshing motion is not too violent as the analytical solution is only valid when $H - H_0$ is small.

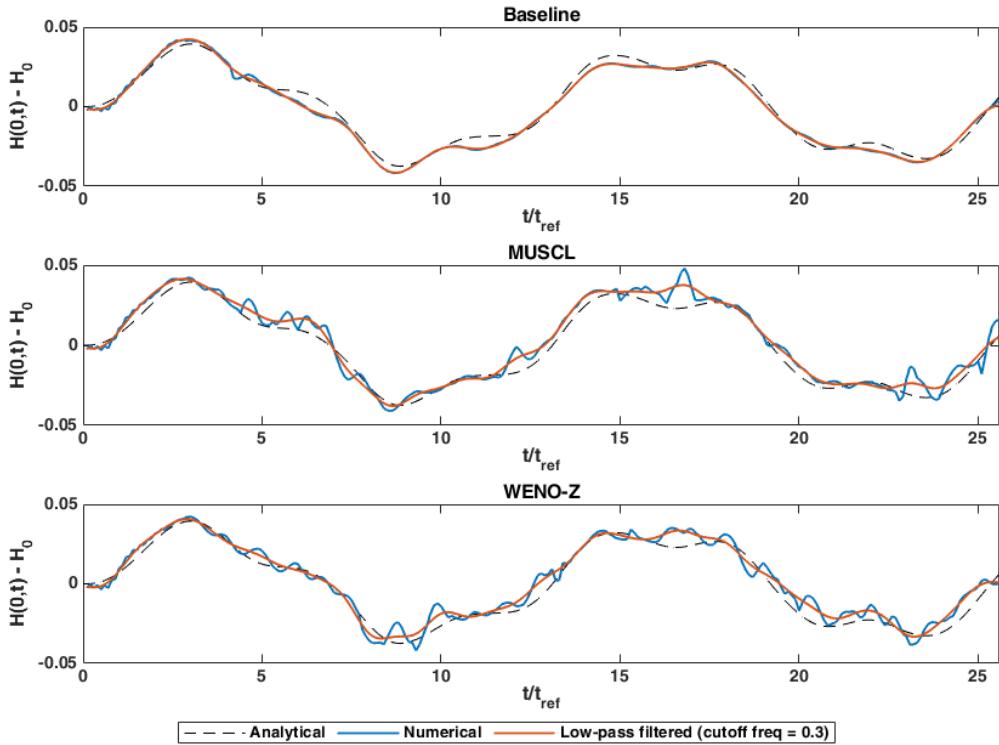


Figure 28: Surface elevation at the left wall ($x = 0$)

The free surface height at $x = 0$ is examined. Figure 28 shows that all three reconstruction schemes correctly simulated the surface elevation, although the solutions of the higher-order schemes are noisier than the first-order scheme. The Fourier spectrum of the system is studied. The analytical solution mainly consists of two peaks, the highest peak locates at the excitation frequency, and one overtone is at the natural frequency. The numerical results correctly captured both peaks, with less than 2% error in the amplitude of the higher peak. The WENO scheme is the most accurate in terms of capturing the amplitudes of the two peaks. At low- to mid-frequency range, the higher-order schemes generate some noise in the signal, which is absent in the baseline scheme because oscillations are damped out quicker. Inspired by the findings from the Fourier spectrum, a low-pass filter is applied to the signals. The cutoff frequency is slightly above the natural frequency, to contain all physical features. The filtered signals match closely with the analytical solution.

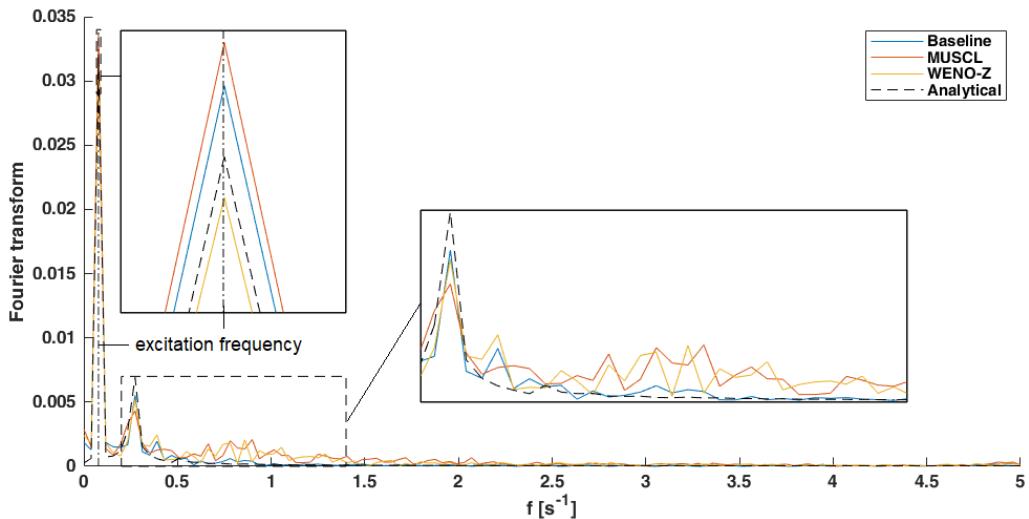


Figure 29: Fourier transform of the surface elevation at the left wall

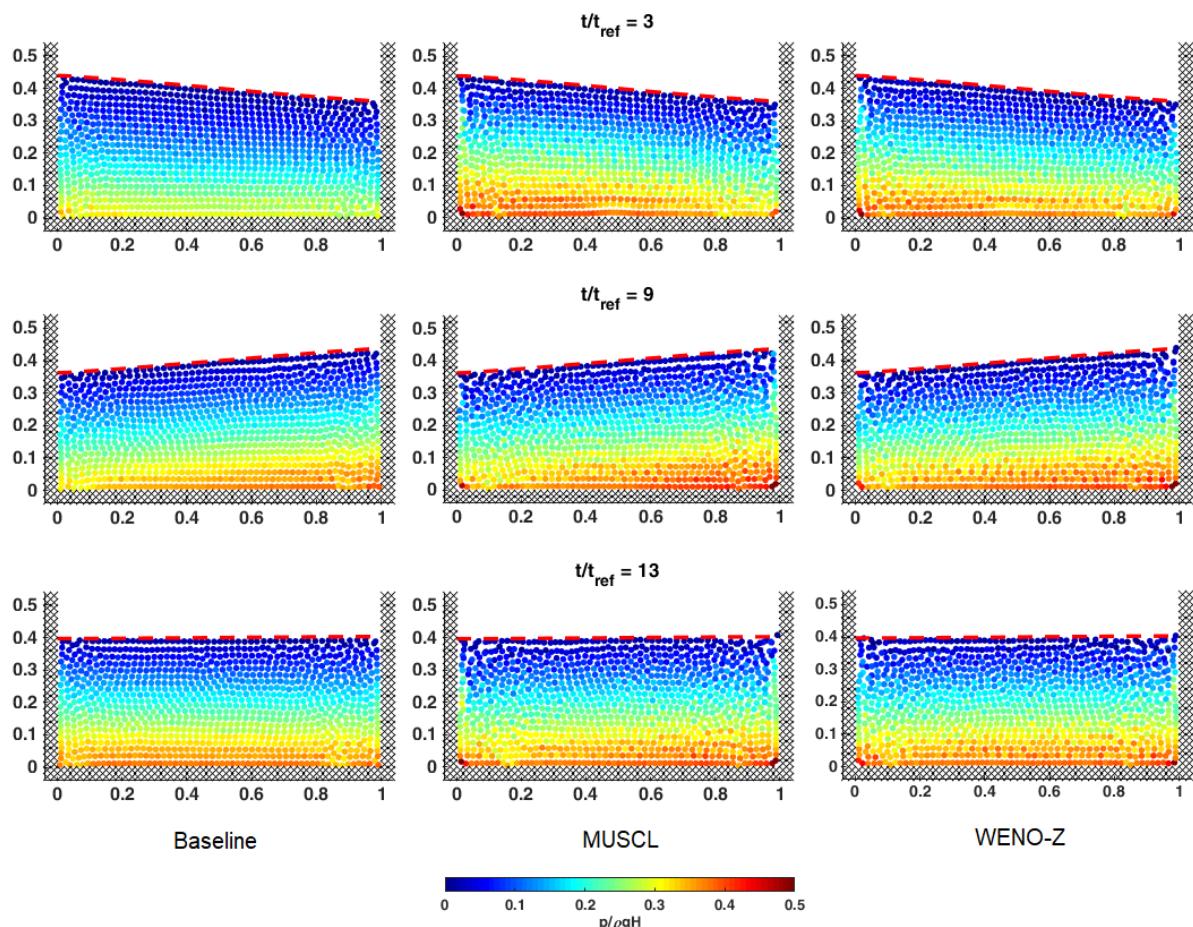


Figure 30: Snapshots of different schemes in linear sloshing case. Red dashed line: analytical solution $H(x, t)$.

6 Conclusions

In this report, the formulation of a Riemann solver based SPH is presented. The focus of our project can be summarised as two points: implicit and low numerical dissipation, and high order of accuracy.

To achieve implicit dissipation, we have implemented a Riemann solver based SPH architecture. Roe's approximate Riemann solver [73], together with Zhang's dissipation limiter [90], are employed because of their documented low dissipation. The problem of nonphysical surface rising is observed in the hydrostatic test case, which leads to our novel artificial diffusion limiter. The mechanism and effects of the limiter are investigated. We have shown that our method produces smooth pressure fields and accurate free surface in all numerical test cases. In the dam-breaking case, our method attains a significant reduction in numerical dissipation comparing to the reference δ -SPH [24]. This is done completely without user input and hence is robust to all cases.

To raise the order of accuracy, we first formulated a stencil construction method based on Taylor series expansion. The method is arbitrarily high-order. We have implemented a third-order version to match the order of accuracy of the chosen WENO-Z scheme. Different formulations of first and second derivative operators are reviewed. With the renormalised derivative scheme [6, 69, 84], our method achieves consistent performance throughout the domain, even when the kernel support becomes incomplete near wall and free surface. Our method can construct the FV-like stencil with up-to third-order accuracy, despite being sensitive to particle disorder.

Based on the stencil constructed, we have implemented and tested three reconstruction schemes, including the first-order Godunov's scheme [33] (baseline), second-order MUSCL [82], and the third-order WENO-Z scheme [7, 9]. All three schemes are stable and accurate in all test cases. Higher-order schemes exhibit less dissipation, but produce slightly noisier flow fields. The accuracy of the schemes are examined using the oscillating droplet benchmark. The baseline and MUSCL schemes are found to have first and second-order accuracy respectively, consistent with the order of the reconstruction scheme. The WENO-Z scheme, on the other hand, achieved a slightly above second-order performance, which is better than most classical SPH methods. Despite being the most accurate among the three, its order of accuracy is still lower than the theoretical value. This is due to the performance of the reconstruction being compromised by the stencil construction method when particles are disordered. Nevertheless, the high-order reconstruction schemes, i.e. MUSCL and WENO, are found to be more computationally efficient than the baseline scheme, therefore, justifies the reason for going high-order.

Finally, we have confirmed that our Riemann solver SPH with all reconstruction schemes can accurately simulate sloshing problem, echoing the title of this report.

6.1 Future works

In terms of the **formulation**, the major bottleneck of our method is the sensitivity of the stencil construction procedure to particle disorder. Therefore, seeking a derivative scheme that is less sensitive to particle disorder will be a good direction to improve on the current method. The Moving Least Square (MLS) approach [3] and the Reproducing Kernel Particle Method (RKPM) [12, 47] demonstrate improved resistance to particle disorder in calculating derivatives, but they are also more complicated and computationally demanding. Furthermore, particle shifting or regularisation strategies, e.g. [16, 46, 76, 89], are becoming increasingly popular in the SPH community. By ‘shifting’ the particles in a way that does not follow their streamlines but satisfy the conservation properties, these methods have the potential of reducing particle disorder and clustering. Furthermore, as the Riemann solver SPH is a relatively new area of research, there are new methods coming up from time to time. Therefore, literature review is extremely important to keep up with the pace of the scientific community.

In terms of **validity of the results**, more test cases should be run to confirm the robustness of the method. Tests with finer particles should also be conducted to investigate the asymptotic region of convergence. To utilise the available computational resources, future studies should consider efficient parallelisation of the code, for example, using C++ and the MPI library, or with GPU acceleration using CUDA [60].

References

- [1] S. Adami, X. Hu, and N. Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.
- [2] M. Antuono, A. Colagrossi, S. Marrone, and D. Molteni. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications*, 181(3):532–549, 2010.
- [3] D. Avesani, M. Dumbser, and A. Bellin. A new class of Moving-Least-Squares WENO–SPH schemes. *Journal of Computational Physics*, 270:278–299, 2014.
- [4] B. Ben Moussa. On the Convergence of SPH Method for Scalar Conservation Laws with Boundary Conditions. *Methods and Applications of Analysis*, 13(1):29–62, 2006.
- [5] J. Bonet and S. Kulasegaram. A simplified approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computation*, 126(2):133–155, 2002.
- [6] J. Bonet and T.-S. L. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180:97–115, 1999.
- [7] R. Borges, M. Carmona, B. Costa, and W. S. Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227(6):3191–3211, 2008.
- [8] B. Buchner. *Green water on ship-type onshore structures*. PhD Thesis, Delft University of Technology, 2002.
- [9] M. Castro, B. Costa, and W. S. Don. High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 230(5):1766–1792, 2011.
- [10] S.-H. Cha. PhD Thesis, Cardiff University, 2002.
- [11] S.-H. Cha and A. P. Whitworth. Implementations and tests of Godunov-type particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 340(1):73–90, 2003.
- [12] J.-S. Chen, W. K. Liu, M. C. Hillman, S.-W. Chi, Y. Lian, and M. A. Bessa. *Reproducing Kernel Particle Method for Solving Partial Differential Equations*, pages 1–44. American Cancer Society, 2017.
- [13] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [14] P. W. Cleary. Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling*, 22:981–993, 1998.

- [15] P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148(1):227–264, 1999.
- [16] A. Colagrossi, B. Bouscasse, M. Antuono, and S. Marrone. Particle packing algorithm for SPH schemes. *Computer Physics Communications*, 183(8):1641–1653, 2012.
- [17] A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2):448–475, 2003.
- [18] A. Connolly, L. Iannucci, R. Hillier, and D. Pope. Second Order Godunov SPH for High Velocity Impact Dynamics. In *3rd South-East European Conference on Computational Mechanicsan ECCOMAS and IACM Special Interest Conference*, Kos Island, Greece, 2013.
- [19] A. J. C. Crespo, M. Gómez-Gesteira, and R. A. Dalrymple. Boundary Conditions Generated by Dynamic Particles in SPH Methods. *Computers, Materials & Continua*, 5(3):173–184, 2007.
- [20] W. Dehnen and H. Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2):1068–1082, 2012.
- [21] J. M. Domínguez, G. Fourtakas, C. Altomare, et al. DualSPHysics: from fluid dynamics to multiphysics problems. *Computational Particle Mechanics*, 2021.
- [22] F. Dubois. *Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to Large Scale Computations*. Nova Science Publishers, Inc, New York, 2001.
- [23] M. Dumbser, M. Käser, V. A. Titarev, and E. F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226(1):204–243, 2007.
- [24] J. R. Eichstädt. *Modelling of sloshing using smoothed particle hydrodynamics*. MSc Thesis, Imperial College London, 2014.
- [25] B. Engquist and S. Osher. Stable and entropy satisfying approximations for transonic flow calculations. *Mathematics of Computation*, 34(149):45–45, 1980.
- [26] M. O. Faltinsen and A. N. Timokha. *Sloshing*. Cambridge University Press, 2009.
- [27] R. Fatehi and M. T. Manzari. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications*, 61(2):482–498, 2011.
- [28] R. Fatehi and M. T. Manzari. A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 67(9):1100–1114, 2011.
- [29] M. Ferrand, D. R. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids*, 71(4):446–472, 2013.

- [30] G. Fourtakas, J. M. Dominguez, R. Vacondio, and B. D. Rogers. Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Computers & Fluids*, 190:346–361, 2019.
- [31] O. Friedrich. Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids. *Journal of Computational Physics*, 144(1):194–212, 1998.
- [32] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 1977.
- [33] S. K. Godunov. Разностный метод численного расчета разрывных решений уравнений гидродинамики (Russian) [A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations]. *Matematicheskii Sbornik*, 47:271–306, 1959.
- [34] M. D. Green. *Sloshing simulations with the smoothed particle hydrodynamics (SPH) method*. PhD Thesis, Imperial College London, 2016.
- [35] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49(3):357–393, 1983.
- [36] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, 71(2):231–303, 1987.
- [37] E. Hewitt and R. E. Hewitt. The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis. *Archive for history of Exact Sciences*, 21(2):129–160, 1979.
- [38] J. Hongbin and D. Xin. On Criterions for Smoothed Particle Hydrodynamics Kernels in Stable Field. *Journal of Computational Physics*, 202(2):699–709, 2005.
- [39] X. Hu and N. Adams. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(2):844–861, 2006.
- [40] X. Hu, S. Jiang, and R. Wang. A MUSCL smoothed particles hydrodynamics for compressible multi-material flows. *Mathematical Methods in the Applied Sciences*, 39(5):1093–1100, 2016.
- [41] S.-I. Inutsuka. Godunov-type SPH. *Memorie della Societa Astronomica Italiana*, 65(4):1027, 1994.
- [42] S.-I. Inutsuka. Reformulation of Smoothed Particle Hydrodynamics with Riemann Solver. *Journal of Computational Physics*, 179(1):238–267, 2002.
- [43] S. Korzilius. *Second derivatives, particle collisions and travelling liquid slugs within smoothed particle hydrodynamics*. PhD Thesis, Technische Universiteit Eindhoven, 2016.
- [44] M. Käser and A. Iske. Ader schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205(2):486–508, 2005.

- [45] S. J. Lind, B. D. Rogers, and P. K. Stansby. Review of smoothed particle hydrodynamics: towards converged lagrangian flow modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2241):20190801, 2020.
- [46] S. J. Lind, R. Xu, P. K. Stansby, and B. D. Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523, 2012.
- [47] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.
- [48] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.
- [49] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013, 1977.
- [50] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani. δ -SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1526–1542, 2011.
- [51] S. Mishra, U. S. Fjordholm, and R. Abgrall. Numerical methods for conservation laws and related equations. Technical report, ETH Zurich.
- [52] J. J. Monaghan. Extrapolating b splines for interpolation. *Journal of Computational Physics*, 60(2):253–262, 1985.
- [53] J. J. Monaghan. Simulating Free Surface Flows with SPH. *Journal of Computational Physics*, 110(2):399–406, 1994.
- [54] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 2005.
- [55] J. J. Monaghan and R. A. Gingold. Shock simulation by the particle method SPH. *Journal of Computational Physics*, 52(2):374–389, 1983.
- [56] J. J. Monaghan and J. B. Kajtar. SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10):1811 – 1820, 2009.
- [57] J. J. Monaghan and J. C. Lattanzio. A refined particle method for astrophysical problems. *Astronomy and astrophysics*, 149:135–143, 1985.
- [58] J. J. Monaghan and A. Rafiee. A simple SPH algorithm for multi-fluid flow with high density ratios. *International Journal for Numerical Methods in Fluids*, 71(5):537–561, 2013.
- [59] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling Low Reynolds Number Incompressible Flows Using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.
- [60] Nvidia Corp. CUDA Toolkit. Available at <https://developer.nvidia.com/cuda-toolkit>. Accessed on 2021-05-31.

- [61] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation*, 38(158):339–339, 1982.
- [62] A. N. Parshikov and S. A. Medin. Smoothed particle hydrodynamics using interparticle contact algorithms. *Journal of Computational Physics*, 180(1):358–382, 2002.
- [63] D. J. Price. *Magnetic fields in Astrophysics*. PhD Thesis, University of Cambridge, 2004.
- [64] D. J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.
- [65] Princeton University. Riemann solvers. Available at <https://princetonuniversity.github.io/Athena-Cversion/AthenaDocsUGRiemann.html>. Accessed on 2021-05-24.
- [66] K. Puri and P. Ramachandran. Approximate Riemann solvers for the Godunov SPH (GSPH). *Journal of Computational Physics*, 270:432–458, 2014.
- [67] N. J. Quinlan, M. Basa, and M. Lastiwka. Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering*, 66(13):2064–2085, 2006.
- [68] A. Rafiee, S. Cummins, M. Rudman, and K. Thiagarajan. Comparative study on the accuracy and stability of SPH schemes in simulating energetic free-surface flows. *European Journal of Mechanics - B/Fluids*, 36:1–16, 2012.
- [69] P. W. Randles and L. D. Libersky. Smoothed Particle Hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375–408, 1996.
- [70] M. Rezavand, C. Zhang, and X. Hu. A weakly compressible SPH method for violent multi-phase flows with high density ratio. *Journal of Computational Physics*, 402:109092, 2020.
- [71] A. Ritter. Die fortpflanzung der wasserwellen (German) [The reproduction of water waves]. *Z. Ver. Dtsch. Ing.*, 36:947–954, 1892.
- [72] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
- [73] P. L. Roe. Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics*, 18(1):337–365, 1986.
- [74] V. V. Rusanov. On difference schemes of third order accuracy for nonlinear hyperbolic systems. *Journal of Computational Physics*, 5(3):507–516, 1970.
- [75] R. D. Ruth. A Canonical Integration Technique. *IEEE Transactions on Nuclear Science*, 30(4):2669–2671, 1983.
- [76] P. N. Sun, A. Colagrossi, S. Marrone, M. Antuono, and A. M. Zhang. Multi-resolution delta-plus-sph with tensile instability control: Towards high reynolds number flows. *Computer Physics Communications*, 224:63–80, 2018.

- [77] P. K. Sweby. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984.
- [78] R. Teyssier and O. Agertz. The Godunov method. Technical report, University of Zurich, 2009.
- [79] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, Berlin, Heidelberg, 1997.
- [80] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the hll-riemann solver. *Shock Waves*, 4:133–155, 1994.
- [81] R. Vacondio, B. D. Rogers, P. K. Stansby, P. Mignosa, and J. Feldman. Variable resolution for SPH: A dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering*, 256:132–148, 2013.
- [82] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *Journal of Computational Physics*, 32(1):101–136, 1979.
- [83] L. Verlet. Computer ”experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967.
- [84] J. P. Vila. On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*, 09(02):161–209, 1999.
- [85] L. Vittoz, G. Oger, M. de Leffe, and D. Le Touzé. Comparisons of weakly-compressible and truly incompressible approaches for viscous flow into a high-order Cartesian-grid finite volume framework. *Journal of Computational Physics: X*, 1:100015, 2019.
- [86] Z.-B. Wang, R. Chen, H. Wang, Q. Liao, X. Zhu, and S.-Z. Li. An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling*, 40(23-24):9625–9655, 2016.
- [87] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [88] C. Zhang. *Smoothed particle hydrodynamics for fluid and solid dynamics*. PhD Thesis, Technical University of Munich (TUM), 2019.
- [89] C. Zhang, X. Y. Hu, and N. A. Adams. A generalized transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 337:216–232, 2017.
- [90] C. Zhang, X. Y. Hu, and N. A. Adams. A weakly compressible SPH method based on a low-dissipation Riemann solver. *Journal of Computational Physics*, 335:605–620, 2017.
- [91] C. Zhang, G. M. Xiang, B. Wang, X. Y. Hu, and N. A. Adams. A weakly compressible SPH method with WENO reconstruction. *Journal of Computational Physics*, 392:1–18, 2019.

A Lid-driven cavity: An example of SPH viscosity model and moving wall boundary condition

Modelling physical viscosity has never been a concern of this project. However, unexpectedly, our code can model physical viscosity, quite accurately indeed. Therefore, I decided to include the results here, for the reference of the readers.

The full momentum equation is

$$\frac{D\mathbf{u}_i}{Dt} = -\frac{1}{\rho_i} \sum_{j \in \mathcal{P}_i} \frac{m_j}{\rho_j} (p_i + p_j + \Pi_{ij}) \nabla W_{ij} + \mathbf{g}, \quad (70)$$

where Π_{ij} is the viscous term. The viscous term involves discretising the Laplacian operator, which, as we have seen in the report, is not very accurate using second derivatives directly. Therefore, viscosity is usually estimated in SPH. While there are many different estimation of the viscous term, we follow the formulation presented by Zhang [88]:

$$\Pi_{ij} = -4\mu \frac{\mathbf{u}_{ij} \cdot \mathbf{r}_{ij}}{r_{ij}^2 + \epsilon}, \quad (71)$$

with μ being the dynamic viscosity, $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ and ϵ is an arbitrary small positive number to prevent division by zero.

We test this model, together with the moving boundary condition in Section 3.5, in the problem below. The baseline scheme is used, with $\Delta x = 0.02$ and $h/\Delta x = 1.5$.

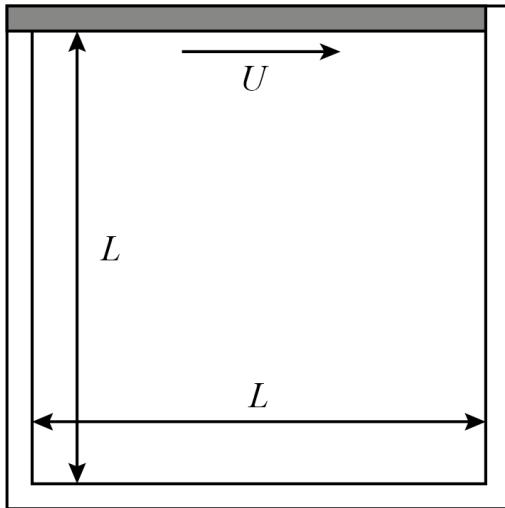


Figure 31: Setup of the lid-driven cavity case

The shaded part of the lid is given a $\mathbf{u}_w = [U, 0]^\top$. Note that the top-right corner of the lid is kept stationary, otherwise fluid will leak from the edge.

The benchmark solution is given by

U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982. doi:10.1016/0021-9991(82)90058-4.

The Reynolds number is defined as

$$Re = \frac{\rho_0 U L}{\mu}. \quad (72)$$

With $U = 1$, $L = 1$, our code shows close agreement with the benchmark solution at $Re = 100$. Cases with higher Re take too long to reach the steady state, and therefore, is left as an exercise for those interested.

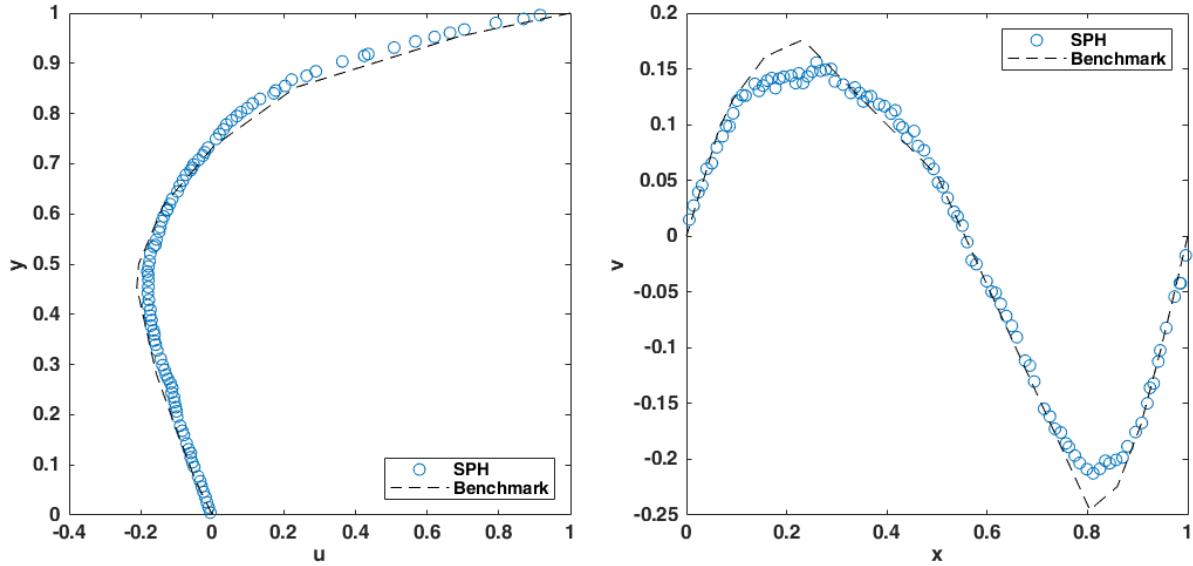


Figure 32: x -velocity along $x = 0.5$ and y -velocity along $y = 0.5$ at steady state

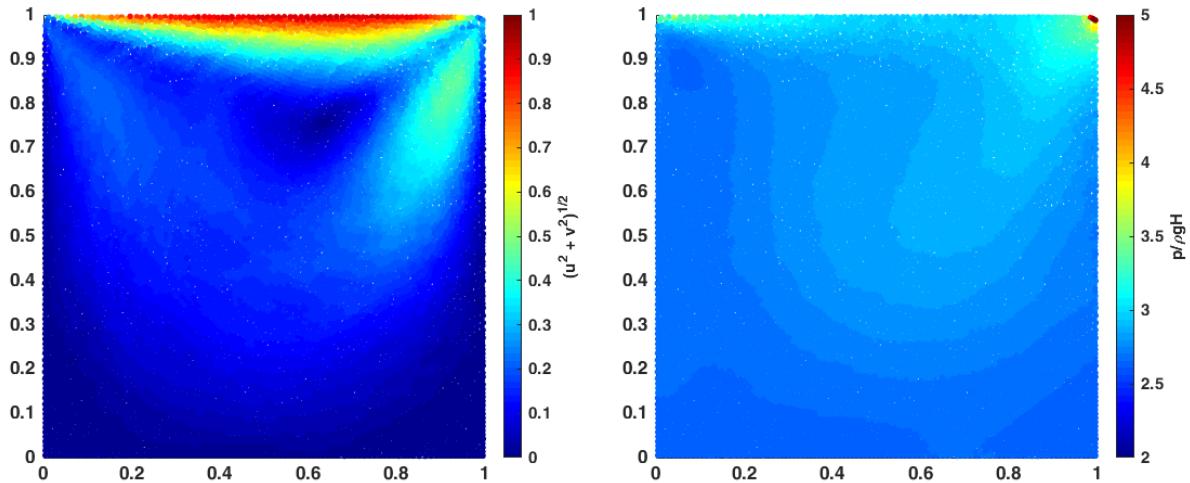


Figure 33: Velocity magnitude and pressure distribution at steady state