

Efficient Eulerian smoothed particle hydrodynamics for fluid dynamics

Zhentong Wang

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Florian Holzapfel
Prüfende der Dissertation:	1. Prof. Dr.-Ing. Oskar J. Haidn
	2. Prof. Dr.-Ing. Pengnan Sun
	3. Priv.-Doz. Dr.-Ing. Xiangyu Hu

Die Dissertation wurde am 16.01.2025 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 27.05.2025 angenommen.

Declaration of Authorship

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Zhentong Wang
July 16, 2025

© Zhentong Wang, 2025
zhentong.wang@tum.de

All rights reserved. No part of this publication may be reproduced, modified, re-written, or distributed in any form or by any means, without the prior written permission of the author.

Released July 16, 2025
Typesetting $\text{\LaTeX}$

Abstract

As one of the meshless methods, smoothed particle hydrodynamics (SPH) can be formulated in both Lagrangian and Eulerian frameworks for flow simulations. Compared to the former, the latter reduces numerical errors and offers higher computational efficiency as its fixed and optimal particle distribution. The Eulerian SPH method is a potential meshless alternative to a traditional Eulerian mesh-based finite volume method (FVM) in computational fluid dynamics (CFD). While previous research has analyzed the differences between these two methods, a rigorous comparison of their performance and computational efficiency is limited by constraints related to the normal direction of interfaces. To address this limitation, the first part of this thesis focuses on developing a certain form of the Eulerian SPH method, which introduces kernel gradient correction to release the directional constraints. This approach allows for the algorithmic equivalence between the Eulerian SPH and FVM. Besides, the mesh-based FVM using a shared solver is implemented within a unified SPH codebase by developing a parser to extract relevant information from mesh files exported in the ".msh" format using the commercial software ICEM. Consequently, a rigorous comparison between these two methods is achieved. Several cases, including compressible and incompressible flows, are tested, demonstrating that while the Eulerian SPH method produces smoother results and, when employing the high-order kernels, achieves a faster convergence rate, it requires more computational effort. Despite achieving algorithmic equivalence between these two methods, notable differences remain in boundary treatment. The second part of this thesis aims to further enhance algorithmic equivalence and pave the way for the strong coupling of these methods within a unified framework. Incorporating the FVM boundary algorithm into the Eulerian SPH method is a critical step to achieving this strong coupling. This is accomplished by allocating additional storage space for particles in the boundaries to meet FVM's boundary treatment, with flux contributions from the boundary surface relying on the zero-order consistency. Numerical examples involving complex geometries and various boundary conditions have been tested, demonstrating the method's accuracy and reliability. To improve the numerical accuracy of the Eulerian SPH method, the third part of this thesis introduces a high-order scheme, i.e., the Weighted Essentially Non-Oscillatory (WENO) reconstruction. The key idea is to construct a four-point stencil along each interacting particle pair and apply WENO reconstruction to determine particle interactions with the Riemann solver. This construction is applied to the Riemann-based Eulerian SPH method with dissipation limiters. Several numerical examples demonstrate that the method can produce smooth results while significantly reducing numerical dissipation. Another approach for improving numerical accuracy is the implementation of high-order kernel methods. This is achieved using kernels characterized by non-monotonic, non-positive regions and high-order vanishing moments for reduced smoothing errors with sufficiently large compact support for decreased integration error. To avoid excessive computational cost from large compact supports, the 4th-order Laguerre-Gauss kernel with an appropriate gradient magnitude is employed. The fourth part of this thesis analyzes the error of the Laguerre-Gauss kernel with

the same compact support as the widely used Wendland kernel. This high-order kernel achieves significantly smaller relaxation residue and truncation errors than other second and fourth-order kernels. Furthermore, a comprehensive set of 2D and 3D benchmark cases validate that the 4th-order Laguerre-Gauss kernel improves the numerical accuracy without compromising the computational efficiency.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to everyone who has contributed to the realization of my dissertation. First and foremost, I am deeply grateful to my doctoral supervisor, *Prof. Dr.-Ing. Oskar Haidn*, and my mentor, *PD Dr.-Ing. Xiangyu Hu*, for their kindness, patience, and guidance in both academic and personal matters. I am incredibly thankful to *PD Hu* for thoroughly advising my work, inspiring me with new ideas, and engaging in insightful discussions. I also greatly appreciate *Prof. Haidn* for providing valuable feedback and inspiring suggestions throughout my studies. Their vision, passion for research, and profound understanding of fluid mechanics have been invaluable, and I consider myself fortunate to have benefited from their expertise.

It has been a pleasure to work alongside my collaborators at the School of Engineering and Design: *Yongchuan Yu*, *Ibraheem Nasser*, *Andrej Sternin*, *Daniel Martínez Sanchis*, *Jianing Liu*, *Chenxi Zhao*, *Xiaojing Tang*, *Mai Ye* and *Hongxin Wang*. I thoroughly enjoyed my time as a Ph.D. student in this group, and it was a privilege to interact with such friendly and talented colleagues. I am also deeply grateful to other members, including *Chi Zhang*, *Bo Zhang*, *Dong Wu*, *Feng Wang* and *Shuaihao Zhang*, for their meaningful and engaging academic discussions.

I consider myself incredibly fortunate to have a loving and supportive family. I am profoundly grateful to my wife, *Ye Tan*, my parents, *Gaobo Wang* and *Xiaoling Lin*, and my siblings. Their unwavering support and encouragement have been instrumental throughout this journey. I am well aware that I could not have reached this milestone without their continuous belief in me. Thank you for standing by my side, for understanding during the most challenging time, and for always believing in me.

Contents

Declaration of Authorship	iii
Abstract	v
Acknowledgements	vii
1 Introduction	1
1.1 Numerical methods for fluid dynamics	1
1.2 Review on SPH	1
1.3 Aims and objectives	3
1.4 Outline	5
2 Governing equations and methods	7
2.1 Governing Equations	7
2.2 Theory and fundamentals of the SPH method	8
2.2.1 Integral interpolation and particle approximation	8
2.2.2 Kernel properties	10
2.2.3 Kernel functions	10
2.3 Compressible and weakly-compressible SPH methods	11
2.3.1 Equation of state	12
2.3.2 Riemann-based discretization equations	12
2.3.3 Time integration	14
2.4 Eulerian SPH with extensions	14
2.4.1 Dissipation limiters	14
2.4.2 Kernel correction matrix	15
2.4.3 Particle relaxation technique	15
2.5 Comparison between Eulerian SPH and FVM	16
2.6 Mesh-based FVM realization in Eulerian SPH	18
2.7 Boundary treatment	18
2.7.1 Wall boundary treatment	18
2.7.2 FVM-equivalent boundary treatment	19
2.8 WENO reconstruction	21
2.9 A fourth-order Laguerre-Gauss kernel	23
2.9.1 Revisit of Laguerre-Wendland kernel	23
2.9.2 Kernel profile	24
2.9.3 Performance in stability	24
2.9.4 Performance in accuracy	26
2.9.5 Convergence analysis	26
2.10 Summary	28

3	Summaries of Publication	29
3.1	FVM realization in Eulerian SPH: A comparative study within unified codebase	29
3.1.1	Summary of the publication	29
3.1.2	Individual novelty of the author	31
3.2	An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework	32
3.2.1	Summary of the publication	32
3.2.2	Individual novelty of the author	32
3.3	An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows	34
3.3.1	Summary of the publication	34
3.3.2	Individual novelty of the author	34
3.4	A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids	35
3.4.1	Summary of the publication	35
3.4.2	Individual novelty of the author	36
4	Conclusions and outlooks	37
4.1	Conclusions	37
4.2	Outlook	38
	Bibliography	39
A	Original journal papers	45
A.1	Paper I	47
A.2	Paper II	65
A.3	Paper III	83
A.4	Paper IV	97

Chapter 1

Introduction

1.1 Numerical methods for fluid dynamics

Fluid dynamics is crucial across industries, from aerospace or other industrial applications to biological fields. Typical examples include the transport plane, atmospheric forecast, rocket engine nozzles, and carotid arteries, as shown in Figure 1.1. With the continuous development of high-performance computers, computational fluid dynamics (CFD) has been recognized as a promising approach to solving practical problems in the real world and also to augment the understanding of many longstanding flow problems [1, 2].

One common approach in CFD is the classical mesh-based methods, such as the Finite Difference Method (FDM) [3], Finite Element Method (FEM) [4], and Finite Volume Method (FVM) [5]. These methods have achieved significant success. However, these face challenges when simulating applications involving significant topological changes. A primary limitation of mesh-based approaches is their reliance on mesh structures, which require computationally intensive processes to generate or regenerate high-quality meshes.

As an alternative, meshless methods, such as the smoothed particle hydrodynamics (SPH) [6, 7], the moving particle semi-implicit (MPS) [8] and the discrete element method (DEM) [9], have gained significant attention over the past few decades. These numerical techniques utilize arbitrarily distributed particles to represent the computational domain and its boundaries, allowing for the construction of governing equations across the entire domain without the need for a predefined mesh. This topological flexibility makes meshless methods particularly advantageous for addressing complex problems and tracking time-dependent field variables at specific material points. Furthermore, these approaches excel in multi-physics problems [10], enabling different materials to be represented by distinct sets of particles, enhancing their versatility for various applications. This thesis focuses on the SPH method, which was chosen for its strong theoretical foundation and well-established principles.

1.2 Review on SPH

The SPH method, initially introduced by Lucy [6] and Gingold and Monaghan [11] for astrophysics, has been widely applied in structural mechanics [12], and other scientific and engineering applications [13, 14, 15]. Extensive reviews have covered its theoretical foundations and principles, as well as diverse applications [11, 16, 17, 18]. Given the key role of SPH in this thesis, a concise overview of its methodology is provided here.

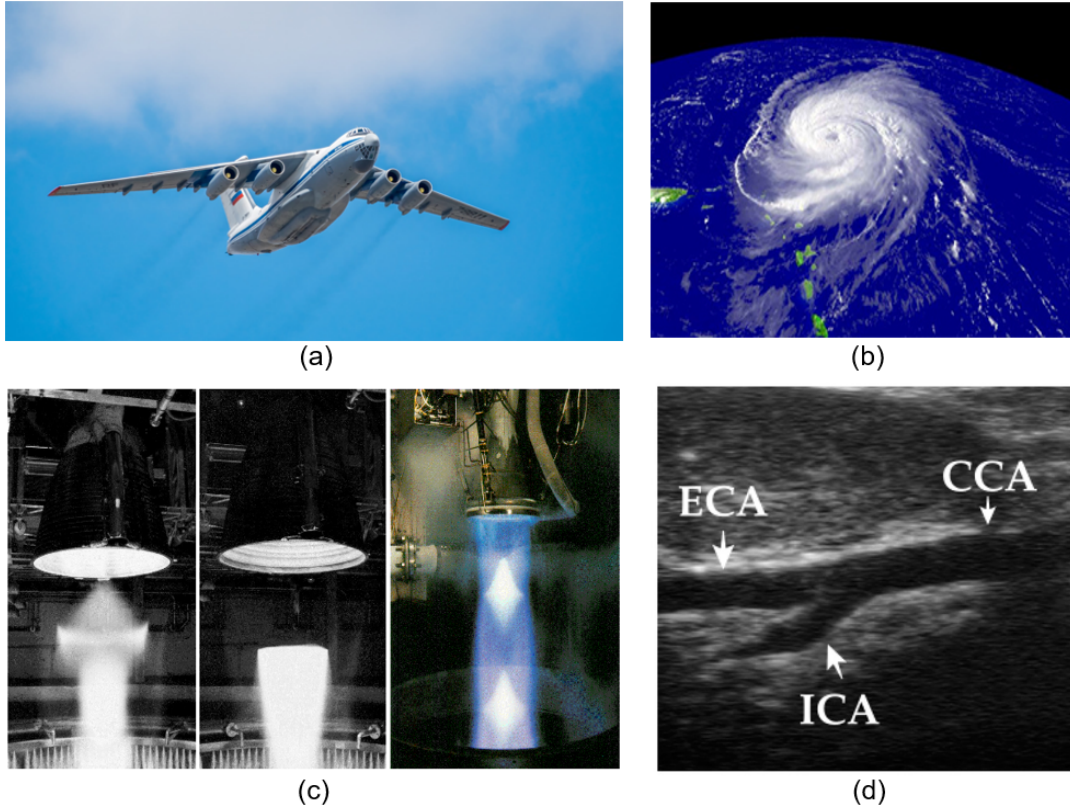


FIGURE 1.1: Typical examples of fluid: (a) Russian II-76 transport plane, (b) Atmospheric forecast, (c) Rocket Engine Nozzles, (d) Carotid artery.

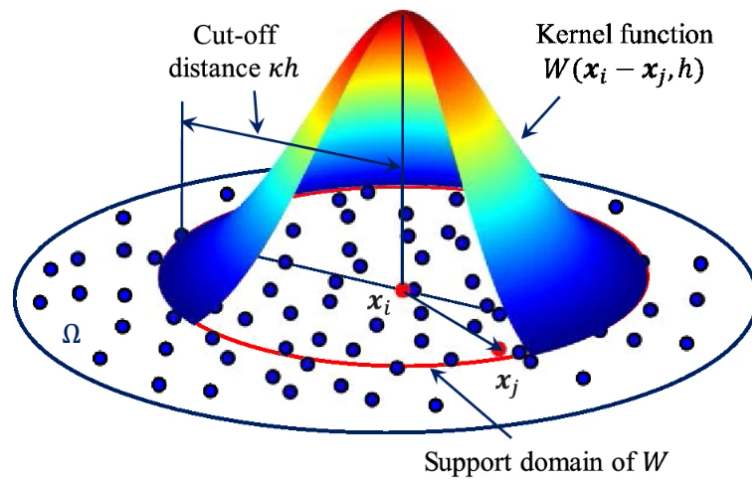


FIGURE 1.2: Schematic view of a 2D SPH particle approximation for particle i using neighbouring particles within the support domain of the kernel function W .

In the SPH method, partial differential equations (PDEs) are formulated in integral form, with the approximation achieved via Gaussian-like kernel function [11, 6]. Once particles discretize the computational domain, its approximation is further conducted, i.e., the integration in the integral representation is replaced by finite summations over values of neighbouring particles within the compact support domain, as illustrated in Figure 1.2 [19]. Specifically, SPH can be applied in both Lagrangian and Eulerian frameworks for flow simulations. In the Lagrangian SPH method, particles move with the flow, offering advantages in cases of topological changes and material interfaces, such as violent free-surface flows, multi-phase interactions, and fluid-structure interaction (FSI) with flexible or rigid bodies [20, 21, 22, 23]. However, such flexibility can result in irregular particle distributions and numerical inaccuracies, necessitating particle regularization techniques [24, 25, 26, 27]. On the other hand, the Eulerian SPH method, with its fixed particle positions, maintains the initial optimal distribution, reducing the large numerical errors [28, 29]. This improves overall accuracy as shown in recent studies [30, 31, 32] and typically offers higher computational efficiency due to the fixed particle configuration [33]. For additional details on SPH theory and fundamentals, see Section 2.2. In the Lagrangian SPH framework, particle approximations are recalculated at each advection time step based on the evolving local particle distribution [34], whereas in the Eulerian SPH framework, these approximations are performed only once [35, 32].

Many researchers have studied the numerical aspects of the Eulerian SPH method, including its accuracy, stability, convergence, and efficiency [17]. Despite significant advancements, several issues within the Eulerian SPH method still require attention. Firstly, in the SPH community, it is recognized that the pairwise particle interaction relying on kernel formulation in SPH discretization can be seen as analogous to the numerical flux at the interface of two computational cells in traditional Eulerian mesh-based methods, such as FVM [36, 26, 37]. Neuhauser [38] provides a detailed analogy, allowing an FVM discretization to utilize the Riemann solver developed for the arbitrary-Lagrangian-Eulerian (ALE) SPH framework within the same software. However, the extent of this analogy between the Eulerian SPH and FVM remains unexplored, and the rigorous comparison between these two methods requires investigation. Secondly, while both FVM and Eulerian SPH methods in Wang et al. [39] utilize the same discretized conservation equations along with unified techniques, the handling of boundary conditions differs between particle-based and mesh-based approaches. Consequently, incorporating the FVM boundary algorithm into the Eulerian SPH method is essential, as it not only aligns the Eulerian SPH method more closely with FVM but also plays a pivotal role in developing a unified SPH framework that enables strong and robust coupling between these two methods. Thirdly, enhancing the numerical accuracy in the Eulerian SPH method remains a critical objective.

1.3 Aims and objectives

The main objective of this thesis is to tackle the challenges mentioned above in the Eulerian SPH method for fluid dynamics.

To enable a rigorous comparison between the FVM and the Eulerian SPH methods, the first task is to present a certain form of the Eulerian SPH method that incorporates a kernel gradient matrix [40], which mitigates constraints related to interface normal directions, achieving algorithmic equivalence with FVM. Additionally, the second task is to implement the traditional FVM formulation within an open-source

SPH library [41]. This is achieved by developing a parser that extracts necessary information from the ".msh" mesh file format generated by ICEM. Finally, a rigorous comparison between these two methods can be conducted by facilitating a shared solver for both methods within a unified codebase. This work is detailed in Paper I [39]

- Zhentong Wang et al. "FVM realization in Eulerian SPH: A comparative study within unified codebase". In: *Computer Physics Communications* (2025), p. 109683

which has been attached in Appendix A.1.

To enhance algorithmic consistency between these two methods, it is essential to incorporate the FVM's boundary algorithm into the Eulerian SPH method. By examining the boundary treatment in FVM, which maintains the zero-order consistency for the cells in the boundaries, we develop a new boundary algorithm tailored for the Eulerian SPH framework. The key idea is that when incorporating the FVM boundary algorithm into the SPH codebase, additional storage is allocated for the particles in the boundaries, ensuring the necessary data is retained to meet FVM's boundary requirements. This work is detailed in Paper II [42]

- Zhentong Wang, Oskar J Haidn, and Xiangyu Hu. "An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework". In: *Computer Physics Communications* 307 (2025), p. 109429

which has been attached in Appendix A.2.

While the methods proposed in Paper I [39] and Paper II [42] introduce the dissipation limiters for enhancing numerical accuracy, further improvements are anticipated by employing a higher-order reconstruction scheme. The key idea involves constructing a 4-point stencil for each interacting particle pair and applying the weighted essentially non-oscillatory (WENO) reconstruction [43] to determine particle interactions using the Riemann solver with dissipation limiters. Although reconstruction has been implemented in the Lagrangian approach, its use within the Eulerian framework remains unexplored. This work is detailed in Paper III [44]

- Zhentong Wang et al. "An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows". In: *Journal of Hydrodynamics* 35.2 (2023), pp. 210–221

which has been attached in Appendix A.3.

Although the high-order WENO scheme in Paper III [44] improves numerical accuracy, it comes at the cost of increased computational effort. An alternative approach for increasing accuracy is to employ high-order kernel methods [45, 26], which utilize kernels with non-monotonic and non-positive regions, as well as high-order vanishing moments, to reduce smoothing errors while providing sufficiently large compact support to minimize integration errors. To address the much computational cost associated with large compact support, a fourth-order Laguerre-Gauss kernel, with the same compact support as the second-order Wendland kernel, has been analyzed and implemented [32]. This kernel provides a substantial improvement in numerical accuracy without compromising computational efficiency. This work is detailed in Paper IV [32].

- Zhentong Wang et al. "A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids". In: *Journal of Computational Physics* (2024), p. 113385

which has been attached in Appendix A.4.

1.4 Outline

The structure of the remainder of this thesis is as follows: Chapters 2.1 to 2.3 introduce the conservation equations of fluid dynamics, the fundamentals of the SPH method, and compressible and weakly-compressible SPH formulations, including the equation of state (EOS), Riemann-based SPH discretization equations, and time integration schemes. Chapter 2.4 elaborates on the Eulerian SPH extension covering dissipation limiters, a kernel correction matrix, and a particle relaxation technique. Chapters 2.5 to 2.6 examine the similarities and differences between the FVM and Eulerian SPH methods, as well as the implementation of FVM in the SPH codebase for the rigorous comparison. Chapter 2.7 presents typical wall boundary treatments and a new FVM-equivalent boundary treatment. Moreover, Chapters 2.8 and 2.9 detail the high-order WENO scheme and the fourth-order Laguerre-Gauss kernel, respectively. Finally, Section 3 summarizes the publications associated with this thesis, and Section 4 provides the conclusions and outlook.

Chapter 2

Governing equations and methods

In this chapter, we briefly summarize the governing equation of fluid dynamics and introduce the theory and fundamentals of the SPH method. Subsequently, we discuss both the compressible and weakly-compressible SPH methods, including EOS, Riemann solvers, and time integration. We also explore the Eulerian SPH extensions, which incorporate dissipation limiters, a kernel correction matrix, and a particle relaxation technique. Following this, we present the rigorous comparison between the Eulerian SPH and FVM, and realize the latter in the former framework. Additionally, we propose a new FVM-equivalent boundary treatment within the Eulerian SPH framework, which aims to enhance algorithmic equivalency with traditional FVM. Finally, we introduce and elaborate on techniques designed to improve numerical accuracy, including the implementation of WENO reconstruction and the fourth-order Laguerre-Gauss kernel.

2.1 Governing Equations

The conservation equation in the Eulerian framework is described as:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (2.1)$$

where $\mathbf{U}$ represents the vector of conserved variables, and $\mathbf{F}(\mathbf{U})$ denotes the corresponding fluxes. Specifically, these can be expressed as:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{bmatrix} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(E + p) \end{bmatrix} + \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(E + p) \end{bmatrix}, \quad (2.2)$$

where u , v , and w denote the velocity components, while ρ and p refer to the density and pressure, respectively. The total energy per unit volume is expressed as $E = \frac{\rho \mathbf{v}^2}{2} + \rho e$, where e represents the internal energy. For viscous flows, a viscous force term is incorporated into the momentum equation as:

$$f^{(v)} = \left(\frac{\partial \mathbf{v}}{\partial t} \right)^{(v)} = \eta \nabla^2 \mathbf{v} \quad (2.3)$$

to the momentum equation. Note that the energy equation is turned off in incompressible flows.

2.2 Theory and fundamentals of the SPH method

Before detailing the governing equations' discretization, the SPH method's fundamentals and theory are first summarized.

2.2.1 Integral interpolation and particle approximation

The SPH method begins its kernel approximation of a function $f(\mathbf{r})$ based on the following fundamental identity:

$$f(\mathbf{r}_i) = \int_{\Omega} f(\mathbf{r}) \delta(\mathbf{r}_i - \mathbf{r}) d\mathbf{r}, \quad (2.4)$$

where $f(\mathbf{r}_i)$ denotes $f(\mathbf{r})$ at a position $\mathbf{r}_i$, $\delta(\mathbf{r}_i - \mathbf{r})$ the Dirac delta function and Ω the volume of the integral domain. When using the Dirac delta function, this integral interpolation ensures the exact reproduction of $f(\mathbf{r})$. However, Eq. (2.4) is unsuitable for constructing numerical models since the Dirac delta function is only defined at a single point. If the Delta function $\delta(\mathbf{r}_i - \mathbf{r})$ is substituted with a smoothing function $W(\mathbf{r}_i - \mathbf{r}, h)$ of finite spatial size h , the kernel approximation [7] of $f(\mathbf{r}_i)$ can be expressed as

$$f(\mathbf{r}_i) \approx \int_{\Omega} f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}), \quad (2.5)$$

where h denotes the smoothing length, which determines the effective area of the smoothing kernel. If the computational domain is discretized into N neighbouring particles (points) $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$, and the value of $f(\mathbf{r}_i)$ is only available at these particles, its approximation can be expressed as:

$$\int_{\Omega} f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) \approx \sum_{j=1}^N f(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h) dV_j, \quad (2.6)$$

where j represents a neighboring particle, and dV_j is the differential volume. When the differential volume element dV_j is replaced with m_j/ρ_j , the approximation becomes:

$$f(\mathbf{r}_i) \approx \int_{\Omega} f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) \approx \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h). \quad (2.7)$$

The summation considers particles within a circle of radius kh centred at particle i with position $\mathbf{r}_i$, as illustrated in Figure 2.1.

Using the SPH kernel approximation from Eq. (2.5), the gradient of a scalar function f can be expressed as:

$$\nabla f(\mathbf{r}_i) \approx \int_{\Omega} \nabla f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}). \quad (2.8)$$

The right-hand side of Eq. (2.8) is integrated by parts and applying Gauss's theorem and can be expressed as

$$\int_{\Omega} \nabla f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) = \int_S f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) \mathbf{n} dS(\mathbf{r}) - \int_{\Omega} f(\mathbf{r}) \nabla W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}), \quad (2.9)$$

where $\mathbf{n}$ represents the unit outward vector normal to the surface S . Given that the smoothing function W typically has compact support, will be noted in Eq. (2.17), the value of W at the boundary of the integral in Eq. (2.9) is zero. Consequently,

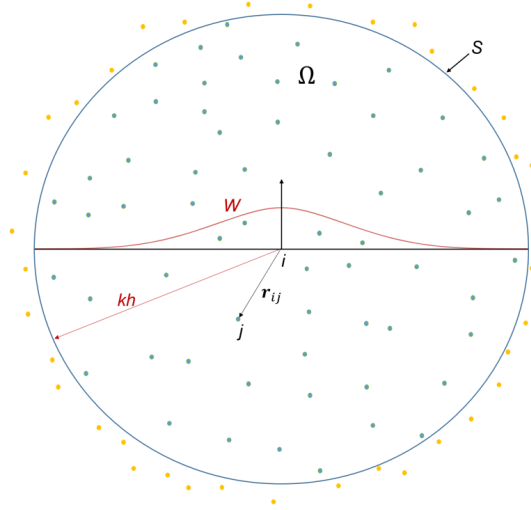


FIGURE 2.1: Sketch of particle approximation of SPH method. Here, κh presents the support length of the smoothing kernel W . The green particles are searched for the particle i , and the orange particles are not the neighbouring particles.

the surface integral on the right-hand side of Eq. (2.8) becomes zero. Therefore, the kernel approximation for the derivatives can be expressed using Eq. (2.8) as

$$\nabla f(\mathbf{r}_i) \approx - \int_{\Omega} f(\mathbf{r}) \nabla W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}). \quad (2.10)$$

Applying particle approximation, described by Eq. (2.5) to Eq. (2.7), Eq. (2.10) can be rewritten as

$$\nabla f(\mathbf{r}_i) \approx - \int_{\Omega} f(\mathbf{r}) \nabla W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) \approx \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{r}_j) \nabla W(\mathbf{r}_i - \mathbf{r}_j, h). \quad (2.11)$$

Here, $\nabla W(\mathbf{r}_i - \mathbf{r}_j, h) = \nabla W_{ij} = \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}$, where $\mathbf{e}_{ij} = -\frac{\nabla W_{ij}}{|\nabla W_{ij}|} = -\mathbf{r}_{ij}/r_{ij}$ with $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ denoting the displacement pointing from particle j to i .

Similarly, the SPH approximation of divergence of a vector function $\mathbf{F}(\mathbf{r})$ at a position $\mathbf{r}_i$ can be derived as

$$\begin{aligned} \nabla \cdot \mathbf{F}(\mathbf{r}_i) &\approx \int_{\Omega} \nabla \cdot \mathbf{F}(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) = \int_{\Omega} \mathbf{F}(\mathbf{r}) \cdot \nabla W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) \\ &\approx \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{F}(\mathbf{r}_j) \cdot \nabla W(\mathbf{r}_i - \mathbf{r}_j, h). \end{aligned} \quad (2.12)$$

The first step in the approximation process is the smoothing approximation, wherein a smooth kernel function replaces the Dirac delta function, and introduces the smoothing error E_s . The second step involves integration by parts, assuming that the kernel function vanishes at the domain boundary. The third step is the integration approximation, achieved by performing a summation over all neighbouring particles, and introduces the integration error E_r . Consequently, the truncation error E_t is obtained by $E_t = E_s + E_r$. To simplify notation, $f(\mathbf{r})$ at positions $\mathbf{r}_i$ and $\mathbf{r}_j$ will henceforth be denoted as f_i and f_j , respectively. Following Ref. [46], Eq. (2.11) can be reformulated

in a strong form as

$$\nabla f_i = \nabla f_i - f_i \nabla 1 \approx \sum_j f_{ij} \nabla W_{ij} V_j, \quad (2.13)$$

with $f_{ij} = f_i - f_j$. Also, the modification in a weak form as

$$\nabla f_i = \nabla f_i + f_i \nabla 1 \approx -2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j, \quad (2.14)$$

with $\bar{f}_{ij} = (f_i + f_j)/2$.

2.2.2 Kernel properties

In the SPH method, a smoothing function is employed to approximate a given function and its derivatives while meeting fundamental requirements. Following Refs. [47, 17, 48], a kernel function adheres to three key conditions:

- The normalization (unity) condition, which requires the kernel function to satisfy:

$$\int_V W(\mathbf{r}_i - \mathbf{r}, h) dV = 1. \quad (2.15)$$

- As the smoothing length approaches zero, the kernel function must exhibit the Dirac delta function property:

$$\lim_{h \rightarrow 0} W(\mathbf{r}_i - \mathbf{r}, h) = \delta(\mathbf{r}_i - \mathbf{r}). \quad (2.16)$$

- The compact support condition, which ensures that:

$$W(\mathbf{r}_i - \mathbf{r}, h) = 0, \quad \text{when} \quad |\mathbf{r}_i - \mathbf{r}| > \kappa h, \quad (2.17)$$

where κ determines the size of the compact support.

2.2.3 Kernel functions

The smoothing kernel plays a crucial role in determining the consistency and accuracy of SPH approximations. Below is a summary of the most widely used smoothing kernels [48, 16, 11, 49, 50, 51]. In the initial SPH study by Lucy [6], a bell-shaped kernel was utilized, given by:

$$W(q, h) = a_d \begin{cases} (1 + 3q)(1 - q)^3 & \text{if } q \leq 1 \\ 0 & \text{if } q > 1 \end{cases}, \quad (2.18)$$

where $q = |\mathbf{r}_i - \mathbf{r}_j|/h$, and a_d represents the normalization coefficient, with d denoting the dimensionality of space. Specifically, $a_1 = 5/(4h)$, $a_2 = 5/(\pi h^2)$, and $a_3 = 105/(16\pi h^3)$.

In another foundational SPH work by Gingold and Monaghan [11], the Gaussian kernel was employed to model non-spherical stars, expressed as:

$$W(q, h) = a_d \exp(-q^2), \quad (2.19)$$

where $a_1 = 1/(\pi^{1/2}h)$, $a_2 = 1/(\pi h^2)$ and $a_3 = 1/(\pi^{3/2}h^3)$. The Gaussian kernel offers high smoothness, even for higher-order derivatives, making it a "gold standard"

for stability and accuracy, particularly with disordered particles. However, the interpolation of the Gaussian kernel extends across the entire computation domain since it does not have a compact support. While the contribution from distant particles quickly diminishes, this characteristic limits its practicality for many applications [52].

Monaghan [51] introduced the cubic spline kernel to enhance computational efficiency by providing a compact support similar in shape to the Gaussian kernel. The expression for this cubic spline kernel is given by:

$$W(q, h) = a_d \begin{cases} \frac{2}{3} - q^2 + \frac{1}{2}q^3 & \text{if } 0 \leq q \leq 1 \\ \frac{1}{6}(2 - q)^3 & \text{if } 1 \leq q \leq 2, \\ 0 & \text{if } 2 \leq q \end{cases}, \quad (2.20)$$

where $\alpha_1 = 1/h$, $\alpha_2 = 15/(7\pi h^2)$ and $\alpha_3 = 3/(2\pi h^3)$. Due to its similarity to the Gaussian kernel but with narrower compact support, this kernel has found extensive use in SPH applications. However, this kernel may lead to the so-called tensile instability in which particles tend to appear pair clumping [53, 54].

Wendland [47] proposed a class of smoothing kernels for SPH method to mitigate problems such as particle clustering and numerical instability. The Wendland kernel is expressed as follows:

$$W(q, h) = a_d \begin{cases} (1 + 2q)(2 - q)^4 & \text{if } 0 \leq q \leq 2 \\ 0 & \text{if } 2 \leq q, \end{cases}, \quad (2.21)$$

where $\alpha_2 = 7/(64\pi h^2)$ and $\alpha_3 = 21/(256\pi h^3)$.

Typically, the accuracy of the SPH method [24, 26] is influenced by the smoothing error E_s and integration error E_r . The smoothing error is determined by the vanishing moments of the kernel function, whereas the integration error arises from the particle summation over neighbouring particles within the cut-off radius. To investigate the smoothing error of these kernels, function $f(\mathbf{r})$ is expanded using Taylor series expansion and is rewritten Eq. (2.5) as

$$\begin{aligned} \int_{\Omega} f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV(\mathbf{r}) = \\ f(\mathbf{r}_i) \int_{\Omega} W(\mathbf{r} - \mathbf{r}_i, h) dV(\mathbf{r}) + \nabla f(\mathbf{r}_i) \int_{\Omega} (\mathbf{r} - \mathbf{r}_i) W(\mathbf{r} - \mathbf{r}_i, h) dV(\mathbf{r}) + \mathcal{O}(h^2). \end{aligned} \quad (2.22)$$

With the normalization condition and even function condition, it is easy to derive

$$\int_{\Omega} f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV = f(\mathbf{r}_i) + \mathcal{O}(h^2), \quad (2.23)$$

indicating that these kernels remain 2nd-order smoothing error.

2.3 Compressible and weakly-compressible SPH methods

In this section, the SPH approach is applied to solve the compressible and incompressible flow problems, and the discretized formulation of the governing equations is presented.

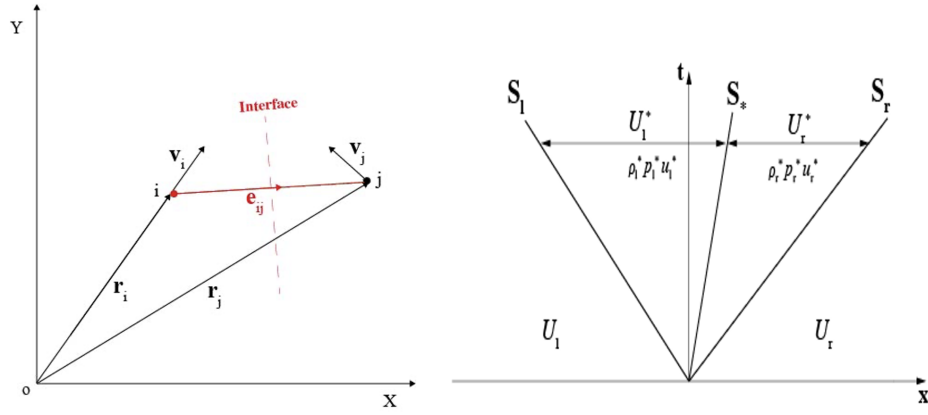


FIGURE 2.2: Construction of Riemann problem along the interacting line of particles i and j where the interface is assumed to pass the middle point and the normal of that is along the interaction line (left panel); Simplified Riemann fan with two intermediate states (right panel).

2.3.1 Equation of state

To close the Eq. (2.1), the EOS is included

$$p = \begin{cases} \rho(\gamma - 1)e & \text{For compressible flows} \\ c^2(\rho - \rho_0) & \text{For incompressible flows} \end{cases} \quad (2.24)$$

where γ represents the heat capacity ratio and ρ_0 is the reference density. Following the ideal gas equation in compressible flows and the weakly-compressible assumption in incompressible flows, the speed of sound is given by

$$c = \begin{cases} \sqrt{\frac{\gamma p}{\rho}} & \text{For compressible flows} \\ 10U_{max} & \text{For incompressible flows} \end{cases} \quad (2.25)$$

where U_{max} represents the maximum velocity in the flow field, ensuring density variation is controlled below 1%.

2.3.2 Riemann-based discretization equations

In the SPH formulation, pairwise particle interactions are represented using a flux-based approach. Specifically, the flux of mass, momentum, and energy across the interface between interacting particles is computed by solving a one-dimensional Riemann problem along the interaction line, denoted as $\mathbf{e}_{ij}$, as shown in Figure 2.2 (left panel). For the Riemann solution, the left (U_l) and right (U_r) states, shown in Figure 2.2 (right panel), of the Riemann problem are defined as [37]

$$\begin{cases} (\rho_l, p_l, u_l, E_l) = (\rho_i, p_i, \mathbf{v}_i \cdot \mathbf{e}_{ij}, E_i) \\ (\rho_r, p_r, u_r, E_r) = (\rho_j, p_j, \mathbf{v}_j \cdot \mathbf{e}_{ij}, E_j) \end{cases} \quad (2.26)$$

assuming that the discontinuity or interface is located at the middle point $\mathbf{r}_{ij} = (\mathbf{r}_i + \mathbf{r}_j)/2$. Then, the Euler equation of Eq. (2.1) can be discretized as [36, 37]

$$\begin{cases} \frac{\partial}{\partial t} (V_i \rho_i) + 2V_i \sum_j V_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (V_i \rho_i \mathbf{v}_i) + 2V_i \sum_j V_j \left[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbb{I} \right] \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (V_i E_i) + 2V_i \sum_j V_j \left[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^* \right] \cdot \nabla W_{ij} = 0 \end{cases} \quad (2.27)$$

Here, V represents the particle volume, $\mathbb{I}$ the identity matrix, and terms $(\cdot)_{E,ij}^*$ the solution of Riemann problem [36]. For the viscous flows, the viscous term in Eq. (2.3) can be discretized as follows:

$$f_i^{(v)} = 2 \sum_j V_j \frac{\eta}{\rho_i} \frac{\mathbf{v}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}}, \quad (2.28)$$

with $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ and r_{ij} denoting the velocity difference and the distance between particles i and j , respectively.

The solution of the Riemann problem results in three waves emanating from the discontinuity, denoted by $(\rho_l^*, u_l^*, p_l^*, E_l^*)$ and $(\rho_r^*, u_r^*, p_r^*, E_r^*)$ as shown in Figure 2.2 (right panel). Two waves, which can be shock or rarefaction waves, travel with the smallest wave speed S_l or the largest wave speed S_r . The middle wave S_* is a contact discontinuity and separates two intermediate states. Toro [55, 56] has proposed the Harten–Lax–van Leer contact (HLLC) solver based on the Harten–Lax–van Leer (HLL) scheme [57] for the more accurate and robust approximation of the Riemann problem for compressible fluid flows. In the HLLC scheme, the wave speeds S_l and S_r estimate for the left and right regions, respectively, are

$$S_l = \tilde{u} - \tilde{c}, S_r = \tilde{u} + \tilde{c}, \quad (2.29)$$

where $\tilde{u}$ and $\tilde{c}$ denoting the average particle and sound speed detailed in Refs. [55, 56]. Then, the intermediate wave speed S_* is calculated as

$$S_* = \frac{\rho_r u_r (S_r - u_r) + \rho_l u_l (u_l - S_l) + p_l - p_r}{\rho_r (S_r - u_r) + \rho_l (u_l - S_l)}. \quad (2.30)$$

Furthermore, other states in the star region can be derived as

$$p^* = p_l + \rho_l (u_l - S_l) (u_l - u^*) = p_r + \rho_r (S_r - u_r) (u^* - u_r), \quad (2.31)$$

$$\mathbf{v}^* = u^* \mathbf{e}_{ij} + \left[\frac{1}{2} (\mathbf{v}_l + \mathbf{v}_r) - \frac{1}{2} (u_l + u_r) \mathbf{e}_{ij} \right], \quad (2.32)$$

$$\rho_{l/r}^* = \rho_{l/r} \frac{(S_{l/r} - u_{l/r})}{(S_{l/r} - u^*)}, \quad (2.33)$$

$$E_{l/r}^* = \frac{(S_{l/r} - u_{l/r}) E_{l/r} - p_{l/r} u_{l/r} + p^* u^*}{S_{l/r} - u^*}, \quad (2.34)$$

with $u^* = S_*$.

For incompressible fluid flows, under the assumption that the intermediate states fulfil $p_l^* = p_r^* = p^*$ and $u_l^* = u_r^* = u^*$, a linearised Riemann solver [58] can be formulated as

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \bar{\rho} \bar{c} (u_l - u_r) \end{cases} \quad (2.35)$$

where $\bar{\rho}$ and $\bar{c}$ represent interface-particle averages.

2.3.3 Time integration

To enhance computational efficiency during the integration of fluid equations, the dual-criteria time-stepping approach is utilized. As described in Refs. [34, 41, 59, 60], this method incorporates two distinct time-step size criteria: the advection criterion, denoted as Δt_{ad} , and formulated as

$$\Delta t_{ad} = \text{CFL}_{ad} \frac{h}{|\mathbf{v}|_{ad}}, \quad (2.36)$$

where the particle advection speed $|\mathbf{v}|_{ad}$ is given by

$$|\mathbf{v}|_{ad} = \max(|\mathbf{v}|_{max}, \max(\frac{v}{h}, U_{ref})), \quad (2.37)$$

and the acoustic criterion, indicated as Δt_{ac} , is defined as

$$\Delta t_{ac} = \text{CFL}_{ac} \frac{h}{|\mathbf{v}|_{max} + c}, \quad (2.38)$$

with $\text{CFL}_{ad} = 0.25$ and $\text{CFL}_{ac} = 0.6$. Here, $|\mathbf{v}|_{max}$, v , and U_{ref} represent the maximum particle fluid speed, kinematic viscosity, and reference velocity, respectively.

2.4 Eulerian SPH with extensions

2.4.1 Dissipation limiters

To address the excessive numerical dissipation in Riemann problems, a variety of high-order techniques have been developed, such as the second-order Monotone Upwind-centered Scheme for Conservation Laws (MUSCL) [61] and the third-order Piecewise Parabolic Method (PPM) [62] [46]. Despite their effectiveness, these approaches tend to impose substantial computational burdens due to the time-consuming data reconstruction processes. To balance computational efficiency and numerical accuracy, an effective dissipation limiter has been integrated into the HLLC Riemann solver for compressible flow simulations [59]. The middle wave speed and pressure in the star region are given by:

$$\begin{cases} u^* = \frac{\rho_l u_l c_l + \rho_r u_r c_r}{\rho_l c_l + \rho_r c_r} + \frac{p_l - p_r}{\rho_l c_l + \rho_r c_r} \beta_{HLLC}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{HLLC} [\rho_r c_r (u^* - u_r) - \rho_l c_l (u_l - u^*)] \end{cases} \quad (2.39)$$

Here, the limiter [59] is defined as:

$$\beta_{HLLC} = \min \left(\eta_{HLLC} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (2.40)$$

For expansion waves ($u_l < u_r$), there is no dissipation, while during compression waves ($u_l > u_r$), the dissipation is modulated by the parameter η_{HLLC} . Numerical experiments suggest $\eta_{HLLC} = 1$ to ensure stability and accuracy for compressible flows in the Eulerian SPH framework.

For incompressible flows, the middle wave speed and pressure in the linearized Riemann solver are calculated as:

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \beta_{linearisd}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{linearisd} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (2.41)$$

where the limiter is defined as

$$\beta_{linearisd} = \min \left(\eta_{linearisd} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (2.42)$$

Based on numerical experiments, $\eta_{linearisd} = 15$ is recommended to maintain stability and accuracy for incompressible flows. Unlike the limiter used in the Lagrangian SPH framework [37], the limiter in velocity term is also incorporated within the Eulerian SPH approach to mitigate excessive dissipation from the convection term in the momentum equation. Additionally, the value of $\eta_{HLLC} = 1$ for compressible flows is different from $\eta_{linearisd} = 15$ for incompressible flows, as the compressible framework has additional numerical dissipation in the energy equation compared to the incompressible framework.

2.4.2 Kernel correction matrix

To further improve numerical accuracy, a kernel gradient correction [59, 40, 63] is also introduced:

$$\mathbf{B}_i = - \left(\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j \right)^{-1}. \quad (2.43)$$

To maintain the conservative property, the kernel correction in a conservative form is defined as:

$$\nabla' W_{ij} = \frac{\mathbf{B}_i + \mathbf{B}_j}{2} \nabla W_{ij}, \quad (2.44)$$

to replace the original kernel gradient ∇W_{ij} in Eq. (2.27).

2.4.3 Particle relaxation technique

It is well established that the distribution of particles plays a critical role in determining numerical accuracy [24]. In the SPH method, particles are initially arranged in a lattice pattern, which often proves inadequate for accurately conforming to the surfaces of complex geometries in real-world applications. Therefore, a straightforward and effective method proposed by Zhu et al. [64] is adopted to generate the body-fitted particle distribution. In this method, the geometry is imported before particle relaxation. The initial particles with lattice distribution are physically moved under a constant background pressure, expressed as

$$\mathbf{a}_i = - \frac{2}{\rho_i} \sum_j V_j \nabla W_{ij}. \quad (2.45)$$

When the acceleration $\mathbf{a}_i$ is zero, particles that are not in the boundaries reach stable positions, satisfying the zero-order consistency condition $\sum_j V_j \nabla W_{ij} = \mathbf{0}$, which indicates that a constant field is being reproduced. Although kernel gradient correction is important for improving numerical accuracy, it has implications for zero-order consistency. Therefore, integrating kernel gradient correction is essential for

TABLE 2.1: Error analysis of SPH approximation for df/dx with a constant function $f = 1$ using the Wendland kernel without and with the kernel gradient correction in the relaxed distribution.

	Without-B	With-B
$L_1(df/dx)$	2.9693×10^{-4}	2.9636×10^{-4}

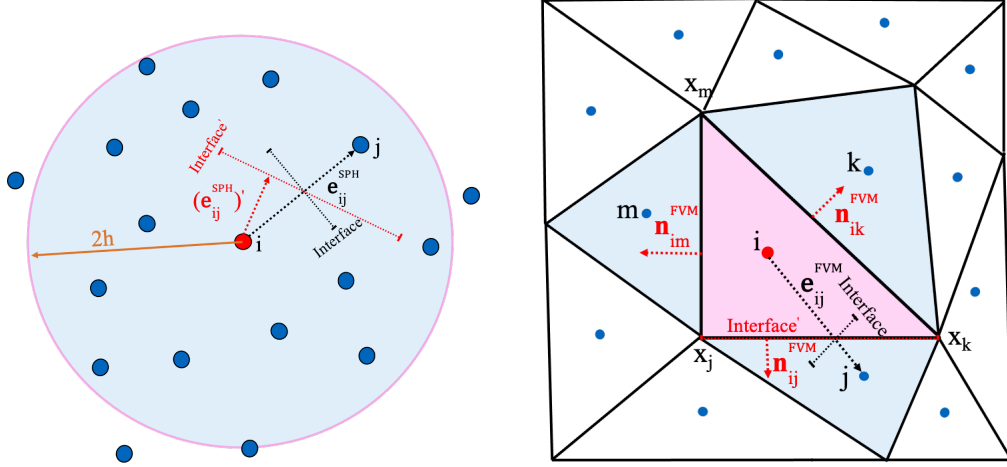


FIGURE 2.3: Flux Integration: Over neighbouring particles in SPH method (left panel); Over neighbouring cells in finite volume method (right panel).

the error analysis related to zero-order consistency. For this error analysis, L_1 normalization is employed, given by:

$$L_1(df/dx) = \frac{1}{N} \sum_i \left| \nabla f_i - 2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j \right|, \quad (2.46)$$

where $2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j$ is the weak form [46] of the SPH approximation, with $\bar{f}_{ij} = \frac{1}{2}(f_i + f_j)$, ensuring conservation as described in Eq. (2.14). In this study, we analyze a circular geometry with a diameter of $D = 2$ and a resolution of $dp = D/20$. A relaxed particle distribution is applied using the Wendland kernel [47], and an error analysis is conducted for df/dx with $f = 1$ to assess zero-order consistency. The L_1 error, presented in Table 2.1, is analyzed under two conditions: without and with kernel gradient correction. The results indicate that applying the kernel gradient correction has minimal impact on the approximation error. Thus, the Eulerian SPH method, coupled with the kernel gradient correction, still maintains zero-order consistency.

2.5 Comparison between Eulerian SPH and FVM

To explore the Eulerian SPH formulation and compare it with FVM, a 2D graphical illustration of both methods is provided in Figure 2.3.

Regarding the similarities between the two approaches, both update the conserved variables by calculating the interacting flux between pairwise particles or cells across all neighbouring elements. In addition, by analogy with the form of the SPH discretization of Eq. (2.27), both methods can be written uniformly [59] as

$$\frac{\partial}{\partial t} (\omega_i \mathbf{U}_i) + \sum_j \mathbf{F}_{ij}(\mathbf{U}) \cdot \mathbf{A}_{ij} = 0. \quad (2.47)$$

Here, $\mathbf{A}_{ij}$ is a vector representing the interface area along the normal direction to the interface.

Furthermore, we can evaluate the differences between the two methods in terms of the Eq. (2.47). The representations of $\mathbf{A}_{ij}$ differ and are indicated respectively as

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{A}_{ij}^{ESPH} = |A_{ij}| \mathbf{n}_{ij}^{ESPH} = 2V_i V_j \nabla W_{ij} = 2V_i V_j \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}^{ESPH} & \text{In Eulerian SPH} \\ \mathbf{A}_{ij}^{FVM} = |A_{ij}| \mathbf{n}_{ij}^{FVM} = S_{ij} \mathbf{n}_{ij}^{FVM} & \text{In FVM} \end{cases}, \quad (2.48)$$

where S_{ij} represents the interface size between cell i and j , with the interface unit normal vector between particles i and j denoted as $\mathbf{n}_{ij}^{SPH}$ for the Eulerian SPH, and $\mathbf{n}_{ij}^{FVM}$ for cells i and j in FVM. In the SPHinXsys library [41], which is an open-source repository utilized in our study, the kernel gradient ∇W_{ij} is represented by storing separately the gradient magnitude, $\frac{\partial W_{ij}}{\partial r_{ij}}$, and the unit displacement vector, $\mathbf{e}_{ij}^{SPH}$. Additionally, the displacement $\mathbf{r}_{ij}^{SPH} = |\mathbf{r}_{ij}^{SPH}| \mathbf{e}_{ij}^{SPH}$ is stored using both the distance $|\mathbf{r}_{ij}^{SPH}|$ and $\mathbf{e}_{ij}^{SPH}$. In FVM, the mesh center is referred to as the centroid, and the displacement between centroids i and j , denoted as $\mathbf{r}_{ij}^{FVM} = |\mathbf{r}_{ij}^{FVM}| \mathbf{e}_{ij}^{FVM}$, is represented along the direction $\mathbf{e}_{ij}^{FVM}$. However, fluxes through the interface are aligned with the unit normal vector $\mathbf{n}_{ij}^{FVM}$, as depicted in Figure 2.3 (right panel). It is important to note that in FVM, the interface unit normal vector $\mathbf{n}_{ij}^{FVM}$ is independent of the direction of $\mathbf{e}_{ij}^{FVM}$. Thus, a significant distinction between Eulerian SPH and FVM is that in Eulerian SPH, the interface normal vector $\mathbf{n}_{ij}^{SPH}$ must align with the displacement direction $\mathbf{e}_{ij}^{SPH}$ [38].

To establish equivalence between these two methods, the kernel correction matrix in Eq. (2.43) is applied to release the constraint and adjust the direction normal to the interface along $(\mathbf{e}_{ij}^{SPH})'$, as depicted in Figure 2.3 (left panel). The modified normal direction is given by

$$(\mathbf{e}_{ij}^{SPH})' = \frac{\nabla' W_{ij}}{|\nabla' W_{ij}|}. \quad (2.49)$$

The interface between two particles in the Eulerian SPH is conceptually similar to the one between cells in FVM, despite being virtual. After releasing the constraint, a virtual mesh structure emerges, which can be considered as generated by the SPH approach. This mesh structure has the property that $(\mathbf{e}_{ij}^{SPH})'$ is orthogonal to the corresponding virtual interface, analogous to the conventional mesh produced by mesh-generation software. Therefore, $(\mathbf{e}_{ij}^{SPH})'$ is analogous to the traditional $\mathbf{n}_{ij}^{FVM}$ in mesh-based FVM.

As shown in Figure 2.3 and Eq. (2.48), the interface size, denoted by $|A_{ij}|$, and neighbouring cells in the FVM are determined using pre-defined grid information, which typically includes three or four neighbours in two or three-dimensional spaces, respectively. On the other hand, in the SPH method, both the interface size and neighbouring particles depend on the kernel gradient and the smoothing length h . Typically, the number of neighbouring particles is around 20 in two dimensions and

approximately 70 in three dimensions, considering a kernel cutoff radius of $2h$, with h defined as $h = 1.3dp$, where dp denotes the initial particle spacing. Moreover, in FVM, each cell represents a completely enclosed control volume that inherently satisfies $\sum_j \mathbf{A}^{FVM}_{ij} = \mathbf{0}$. Conversely, in the Eulerian SPH method, due to relaxation residues, particles not in the boundaries result in $\sum_j \mathbf{A}^{ESPH}_{ij} \approx \mathbf{0}$, introducing consistency errors [38], as discussed in Section 2.4.3.

2.6 Mesh-based FVM realization in Eulerian SPH

This section presents the implementation of the mesh-based FVM within the Riemann-based Eulerian SPH framework. To achieve this, a parser is developed to extract essential data from an external mesh file produced by ICEM, a commercial software tool, thereby facilitating the integration of FVM with the SPH codebase. The mesh file provides direct access to information on node positions and mesh topology. Utilizing this data, other necessary properties, such as interface size, interface unit normal vectors, and distances between centroids, can be computed.

Based on the relation between two methods in Eq. (2.48), it can be deduced that

$$\frac{\partial W_{ij}}{\partial r_{ij}} = \frac{S_{ij}}{2V_i V_j}. \quad (2.50)$$

To adapt the traditional mesh-based FVM within the SPH framework, following the SPHinXsys data structure, we represent the FVM's storage analogous to the SPH method. This includes two components: the interface unit normal vector $\mathbf{n}^{FVM}_{ij}$ and the magnitude of kernel gradient $\frac{S_{ij}}{2V_i V_j}$ where $\mathbf{n}^{FVM}_{ij}$ and S_{ij} are calculated from the mesh information. It is crucial to note that $\frac{S_{ij}}{2V_i V_j}$ in FVM is labeled as the magnitude of the kernel gradient but is solely influenced by the interface size, independent of the kernel itself. This ensures data structure consistency and allows compatibility in the SPH codebase, thus enabling the realization of both mesh-based FVM and SPH methods. Additionally, we store r^{FVM}_{ij} within the storage of r_{ij} , which is used in calculating the viscous force in Eq. (2.28). With this equivalence between the Eulerian SPH and FVM, as outlined in Section 2.4.2, the same Riemann solver and dissipation limiters from Eqs. (2.39) and (2.41) can be consistently applied across both methods.

2.7 Boundary treatment

2.7.1 Wall boundary treatment

Handling wall-boundary conditions is one of the most challenging aspects of the SPH method, and numerous approaches have been developed to address this problem [65, 66]. Below, we provide a brief summary of three techniques from the literature for treating wall-boundary conditions:

- The ghost particle technique, illustrated in Figure 2.4 (left panel), is extensively employed in particle-based methods to enforce boundary conditions. This approach involves introducing artificial particles with specified physical properties, such as pressure and velocity, to occupy the empty space within the kernel support near boundaries. This approach efficiently mitigates nonphysical behaviours. However, implementing it in three-dimensional complex geometries

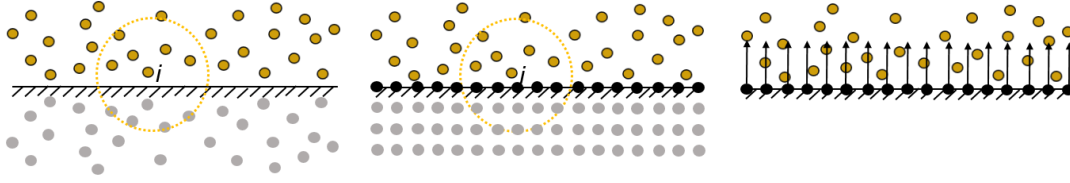


FIGURE 2.4: Solid boundary treatment: ghost particles (left panel), dummy particles (middle panel), repulsive force (right panel).

remains challenging, primarily due to difficulties associated with accurately positioning ghost particles.

- The dummy particle method, developed by Adami et al. [65], represents boundaries with multiple layers of fixed particles, spaced identically to the initial fluid particles to ensure proper convolution of the surrounding fluid particles, as illustrated in Figure 2.4 (middle panel). These dummy particles are governed by the same continuity and momentum equations as the fluid particles.
- Monaghan and Kajtar [66] introduced a method utilizing a single layer of fixed particles interacting with fluid particles via radial forces to enforce solid wall boundary conditions, illustrated in Figure 2.4 (right panel). This approach is computationally efficient and straightforward to apply, particularly for complex boundaries. However, it struggles to keep stationary particles stable under hydrostatic conditions.

Beyond these boundary treatments, several novel methods have been proposed, such as the unified semi-analytical boundary approach [67]. Specifically, in the Riemann-based SPH framework, the fluid-wall interaction is modelled by solving a one-sided Riemann problem [68], which is one of the boundary methods implemented in our work, along the direction normal to the boundary. Wall particles are incorporated as predefined elements of the contact configuration, effectively complementing the inner fluid particle configuration. In the one-sided Riemann problem, the left state is determined by the fluid particle that corresponds to the normal direction of the local boundary, and defined as follows:

$$(\rho_l, u_l, p_l, E_l) = (\rho_f, -\mathbf{n}_w \cdot \mathbf{v}_f, p_f, E_f), \quad (2.51)$$

where the subscript f refers to fluid particles, and $\mathbf{n}_w$ indicates the direction normal to the wall, as depicted in Figure 2.5. The right-state velocity u_r is obtained based on the physical boundary conditions of the wall as follows:

$$u_r = 2u_w - u_l, \quad (2.52)$$

where u_w is the wall velocity. Besides, the pressure [65] in the right state is given by:

$$p_r = p_l + \rho \mathbf{g} \cdot \mathbf{r}_{fw}, \quad (2.53)$$

where $\mathbf{g}$ is the gravity force, $\mathbf{r}_{fw} = \mathbf{r}_w - \mathbf{r}_f$, and the right-state density is obtained by applying the artificial equation of state.

2.7.2 FVM-equivalent boundary treatment

As previously mentioned, both SPH and FVM methods can be expressed as illustrated in Eq. (2.47). To improve clarity, we categorize all cells in FVM and particles

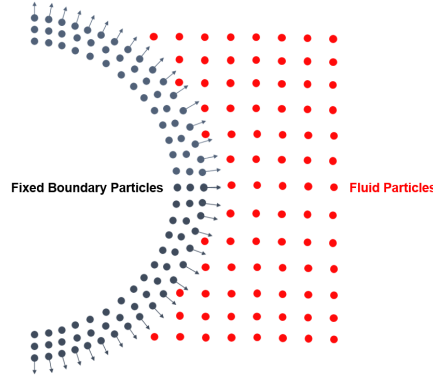


FIGURE 2.5: Sketch of fluid particles interacting with fixed wall boundary particles along the wall normal direction through the one-side Riemann problem.

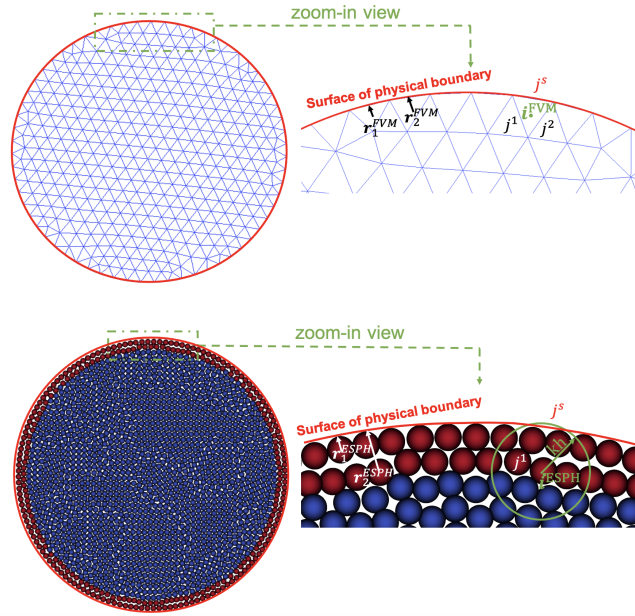


FIGURE 2.6: Mesh (top panel) and particle (bottom panel) distributions and their zoom-in views in FVM and Eulerian SPH, respectively, for a circular geometry.

in SPH into two groups: those in the boundaries and those not in the boundaries. For cells or particles in the boundaries, we distinguish the flux contributions into two components: from neighbouring cells/particles (denoted by j^N) and from the boundary surface (denoted by j^S). For the cells or particles not in the boundaries, j^N represents all neighbouring particles within the whole compact support or all neighbouring cells.

In FVM, cells not in the boundaries completely form closed control volumes, as described in Refs. [59, 38]. These volumes inherently and strictly meet the condition $\sum_j \mathbf{A}_{ij}^{FVM} = \mathbf{0}$, representing the zero-order consistency. For cells in the boundaries, such as cell i^{FVM} shown in Figure 2.6 (top panel), the control volume requires to incorporate the boundary surface to ensure closure. Consequently, all cells adhere to the zero-order consistency. For a better understanding, the zero-order consistency for cells in the boundaries can be reformulated as

$$\sum_j \mathbf{A}_{ij}^{FVM} = \sum_{j^N} \mathbf{A}_{ij^N}^{FVM} + \mathbf{A}_{ij^S}^{FVM} = \mathbf{0}. \quad (2.54)$$

In a two-dimensional unstructured mesh with $N = 1$ or 2 , as depicted by j^1 and j^2 in Figure 2.6 (top panel), the flux contribution from neighbouring cells is represented by $\sum_{j^N} \mathbf{A}_{ij^N}^{FVM} = \sum_{j^N} S_{ij^N} \mathbf{n}_{ij^N}^{FVM}$. Similarly, the flux contribution from the boundary surface is given by $\mathbf{A}_{ijs}^{FVM} = S_{ijs} \mathbf{n}_{ijs}^{FVM}$. The boundary term $\mathbf{A}_{ijs}^{FVM}$ can be directly computed within the FVM framework since both the interface size S_{ijs} and the unit normal vector $\mathbf{n}_{ijs}^{FVM}$ are from the mesh file.

In the Eulerian SPH, the approximation is based on locating neighbouring particles within the compact support [11]. Once the particle relaxation process in Eq. (2.45) is completed, it is assumed that particles not in the boundaries satisfy the zero-order consistency. However, particles in the boundaries, such as particle i^{ESPH} shown in Figure 2.6 (bottom panel), have an insufficient number of neighbours, leading to a failure in fulfilling the zero-order consistency condition. To address this issue, similar to Eq. (2.54), strict zero-order consistency is enforced in the Eulerian SPH method, formulated as

$$\sum_j \mathbf{A}_{ij}^{ESPH} = \sum_{j^N} \mathbf{A}_{ij^N}^{ESPH} + \mathbf{A}_{ijs}^{ESPH} = \mathbf{0}, \quad (2.55)$$

where $\sum_{j^N} \mathbf{A}_{ij^N}^{ESPH} = 2V_i V_{j^N} \sum_{j^N} \nabla W_{ij^N}$ represents the flux contribution from all neighbouring particles. Additionally, $\mathbf{A}_{ijs}^{ESPH}$ denotes the flux contribution from the boundary surface. Unlike $\mathbf{A}_{ijs}^{FVM}$ in the FVM, calculating $\mathbf{A}_{ijs}^{ESPH}$ directly is challenging due to the difficulty of obtaining the boundary surface size within the Eulerian SPH method. Consequently, $\mathbf{A}_{ijs}^{ESPH}$ is computed through Eq. (2.55) as presented below:

$$\mathbf{A}_{ijs}^{ESPH} = - \sum_{j^N} \mathbf{A}_{ij^N}^{ESPH}. \quad (2.56)$$

Thus, particles not in the boundaries achieve approximate zero-order consistency with minimal error, as discussed in Section 2.4.3, while particles in the boundaries strictly adhere to the zero-order consistency. With this consistency satisfied, the flux contribution from the boundary surface is obtained accurately, thus avoiding instabilities like the one-sided flux problem.

2.8 WENO reconstruction

The WENO reconstruction is employed effectively for improving the numerical accuracy. Following Ref. [43], a 4-point stencil is constructed with an incremental size defined along each interacting particle pair, as shown in Figure 2.7. For each stencil, the mid-point variable is expressed as

$$q_{1/2} = \sum_{k=1}^3 w_k q_{1/2}^{(k)}, \quad (2.57)$$

where

$$\begin{aligned} q_{1/2}^{(1)} &= \frac{1}{2}q_0 + \frac{1}{2}q_1, & q_{1/2}^{(2)} &= -\frac{1}{2}q_{-1} + \frac{3}{2}q_0, \\ q_{1/2}^{(3)} &= \frac{1}{3}q_0 + \frac{5}{6}q_1 - \frac{1}{6}q_2 \end{aligned} \quad (2.58)$$

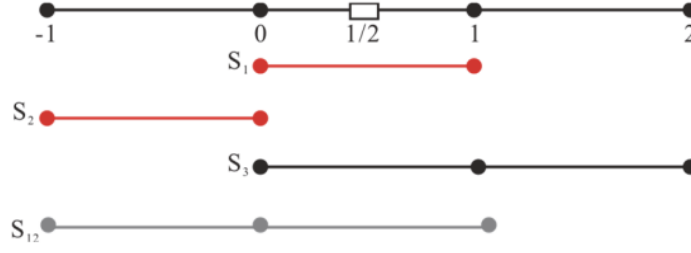


FIGURE 2.7: Four-point stencil and candidate stencils for the incremental-stencil WENO reconstruction.

are the candidate stencils and w_k represent the corresponding non-linear weights calculated as

$$w_k = \frac{\alpha_k}{\sum_{s=1}^3 \alpha_s}, \quad (2.59)$$

with

$$\begin{aligned} \alpha_1 &= d_1 \left(C + \frac{\tau_4}{\beta_1 + \epsilon} \frac{\tau_4}{\beta_{12} + \epsilon} \right), \alpha_2 = d_2 \left(C + \frac{\tau_4}{\beta_2 + \epsilon} \frac{\tau_4}{\beta_{12} + \epsilon} \right), \\ \alpha_3 &= d_3 \left(C + \frac{\tau_4}{\beta_3 + \epsilon} \right), \end{aligned} \quad (2.60)$$

where $d_1 = \frac{1}{3}$, $d_2 = \frac{1}{6}$, $d_3 = \frac{1}{2}$, $\epsilon = 10^{-6}$ and C is a physically motivated scale-separation parameter. Besides, a global reference smoothness indicator τ_4 given by

$$\begin{aligned} \tau_4 &= [q_{-1}(547q_{-1} - 2522q_0 + 1922q_1 - 494q_2) \\ &\quad + q_0(3443q_0 - 5966q_1 + 1602q_2) \\ &\quad + q_1(2843q_1 - 1642q_2) + 267q_2q_2] / 240 \end{aligned} \quad (2.61)$$

is introduced to improve the overall accuracy [69, 70]. Note that the smoothness indicators in Eq. (2.60) are given as:

$$\begin{aligned} \beta_1 &= (q_1 - q_0)^2, \quad \beta_2 = (q_0 - q_{-1})^2, \\ \beta_{12} &= \frac{1}{4}(q_{-1} - q_1)^2 + \frac{13}{12}(q_{-1} - 2q_0 + q_1)^2, \\ \beta_3 &= \frac{13}{12}(q_0 - 2q_1 + q_2)^2 + \frac{1}{4}(3q_0 - 4q_1 + q_2)^2. \end{aligned} \quad (2.62)$$

For each interacting particle pair i and j , the variable values at the stencil point are defined as:

$$\begin{aligned} q_{-1} &= \Phi_i - \nabla \Phi_i \cdot \mathbf{r}_{ij}, \quad q_0 = \Phi_i, \quad q_1 = \Phi_j, \\ q_2 &= \Phi_j + \nabla \Phi_j \cdot \mathbf{r}_{ij}, \end{aligned} \quad (2.63)$$

where

$$\nabla \Phi_i = \sum_j \frac{m_j}{\rho_j} (\Phi_j - \Phi_i) \nabla_i W_{ij} \quad (2.64)$$

is the gradient of primitive value Φ_j with a standard SPH approximation. Afterwards, the left state of the Riemann problem is $\Phi_L = q_{1/2}$ and the corresponding

TABLE 2.2: Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the Wendland and Laguerre-Wendland kernels with different κ without the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Wendland	1.3	0.034	0.123	0.259	0.400
Laguerre-Wendland		0.070	0.673	1.880	3.506
Wendland	2.0	0.014	0.073	0.027	0.150
Laguerre-Wendland		0.010	0.018	0.028	0.046
Wendland	2.5	0.018	0.078	0.032	0.161
Laguerre-Wendland		0.005	0.009	0.017	0.030

right state Φ_R can be calculated from Eq. (2.57) with the mirrored stencil defined as:

$$\begin{aligned} q_{-1} &= \Phi_j + \nabla \Phi_j \cdot \mathbf{r}_{ij}, & q_0 &= \Phi_j, & q_1 &= \Phi_i, \\ q_2 &= \Phi_i - \nabla \Phi_i \cdot \mathbf{r}_{ij}. \end{aligned} \quad (2.65)$$

2.9 A fourth-order Laguerre-Gauss kernel

2.9.1 Revisit of Laguerre-Wendland kernel

Following Litvinov et al. [26], both the Laguerre-Wendland and Laguerre-Gauss kernels exhibit a second vanishing moment and a fourth-order smoothing error. The Laguerre-Wendland kernel [26] is defined as:

$$w^{LW}(s) = C_{LW}(1 - \epsilon^2 s^2 + \epsilon^4 s^4/6)(1 - s)_+^6(1 + 6s + 35/3s^2), \quad (2.66)$$

where $(\cdot)_+ = \max(0, \cdot)$, $s = |\mathbf{r}|/h$, with the normalization constant $C_{LW} \approx 7.591$ and the scaling factor $\epsilon \approx 2.821$ for the range $s \in [0, 1]$. The Laguerre-Wendland kernel has not seen extensive practical use due to its significantly larger integration error than the widely employed second-order Wendland kernel. In this work, we revisit the Laguerre-Wendland kernel to explore the reason for the large integration error. Analysis and comparison of kernel profiles and their derivatives, as shown in Figure 2.8, reveal that the gradient magnitude of the Laguerre-Wendland kernel is significantly larger than that of the Wendland kernel. Thus, we utilize various κ , including $\kappa = 1.3, 2.0$, and 2.5 , representing different total numbers of neighbouring particles to investigate the integration error. To facilitate quantitative analysis, L_1 -norm and L_{∞} -norm, i.e., $L_{\infty}(df/dx) = \max_i \left| \nabla f_i + 2 \sum_j \tilde{f}_{ij} \nabla W_{ij} V_j \right|$, errors are given.

This study employs a circular domain with a diameter of $D = 2$ with the resolution of $dp = 1/60$. We assess the SPH approximation of the derivatives for two distinct functions: a constant function $f = 1$ and a trigonometric function $f = \sin(2\pi x)$. These evaluations are performed using both Wendland and Laguerre-Wendland kernels, with and without the kernel correction matrix. The particle distributions resulting from the different kernels are depicted in Figure 2.9. Truncation errors corresponding to varying values of κ are listed in Tables 2.2 and 2.3. The analysis indicates that truncation errors for the Laguerre-Wendland kernel exhibit similar characteristics to those of the Wendland kernel. Specifically, the errors decrease for $\kappa \leq 2$ and stabilize at $\kappa = 2.5$. This finding suggests that the smoothing error dominates for $\kappa \geq 2$, while the integration error becomes dominant at $\kappa = 1.3$.

TABLE 2.3: Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the Wendland and Laguerre-Wendland kernels with different κ with the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\text{infinity}}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Wendland	1.3	0.039	0.052	0.209	0.187
Laguerre-Wendland		0.077	0.067	2.334	2.319
Wendland	2.0	0.014	0.042	0.026	0.067
Laguerre-Wendland		0.011	0.007	0.029	0.028
Wendland	2.5	0.018	0.043	0.033	0.068
Laguerre-Wendland		0.005	0.003	0.017	0.015

When the smoothing error is dominant ($\kappa \geq 2$), the truncation error of the Laguerre-Wendland kernel is smaller than the Wendland kernel, as shown in Table 2.3. This outcome corresponds well with the smoothing order characteristics of these kernels.

2.9.2 Kernel profile

The Laguerre-Gauss kernel, another fourth-order kernel illustrated in Figure 2.8, presents a smaller gradient magnitude than the Laguerre-Wendland kernel. This characteristic makes it a promising alternative for reducing significant integration errors. The Laguerre-Gauss kernel is defined as given in [26]:

$$w(s) = \pi^{-1} L_1^{(1)}(s^2) e^{-s^2}, \quad (2.67)$$

where $L_1^{(1)}$ represents the generalized Laguerre polynomial, characterized by a non-negative Fourier transform and vanishing second-order moment. For practical implementation, the kernel's domain is truncated to $2h$, and the truncated Laguerre-Gauss kernel is given by:

$$w(s) = \alpha_d \begin{cases} (1 - \frac{s^2}{2} + \frac{s^4}{6}) e^{-s^2} & \text{if } 0 \leq s \leq 2 \\ 0 & \text{if } s > 2 \end{cases}, \quad (2.68)$$

with $\alpha_1 = 8/(5 \times \sqrt{\pi})$, $\alpha_2 = 3/\pi$, and $\alpha_3 = 8/\pi^{3/2}$. The truncated Laguerre-Gauss kernel and its derivative are compared with the Wendland and Laguerre-Wendland kernels in Figure 2.8 where the gradient magnitude of the truncated Laguerre-Gauss kernel is larger than that of the Wendland kernel but smaller than that of the Laguerre-Wendland kernel, and the kernel's function value at $2h$ is close to zero. In simulations, a compact support of $2h$ with $h = 1.3dp$ ensures comparable computational efficiency for the truncated Laguerre-Gauss and Wendland kernels.

2.9.3 Performance in stability

To investigate the properties of the truncated Laguerre-Gauss kernel, we start by analyzing the relaxed particle distributions generated using the Wendland, Laguerre-Wendland, and truncated Laguerre-Gauss kernels within the aforementioned circular domain. Figure 2.9 depicts the variation of relaxation residue ranging from 3.4×10^{-6} to 4.0×10^{-2} using the Wendland, Laguerre-Wendland, and truncated Laguerre-Gauss kernels in the final iteration of the relaxation process. Although it was anticipated that the truncated Laguerre-Gauss kernel, given its resemblance to

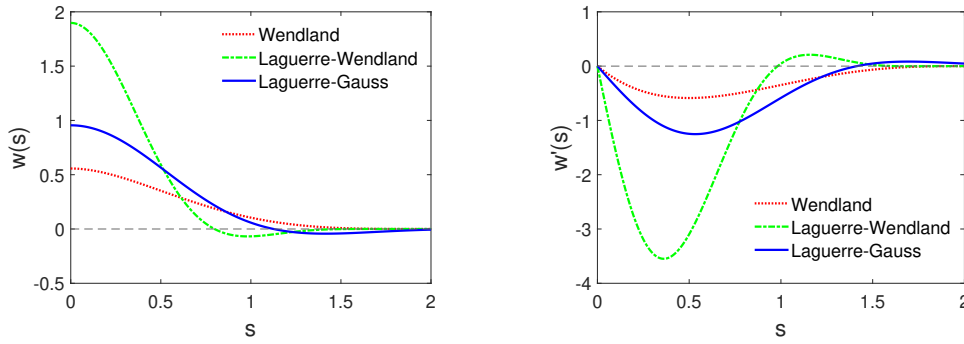


FIGURE 2.8: Laguerre-Gauss kernel and the comparison with Wendland as well as Laguerre-Wendland kernels with compact support as $2h$: Kernel functions (left panel) and their derivatives (right panel).

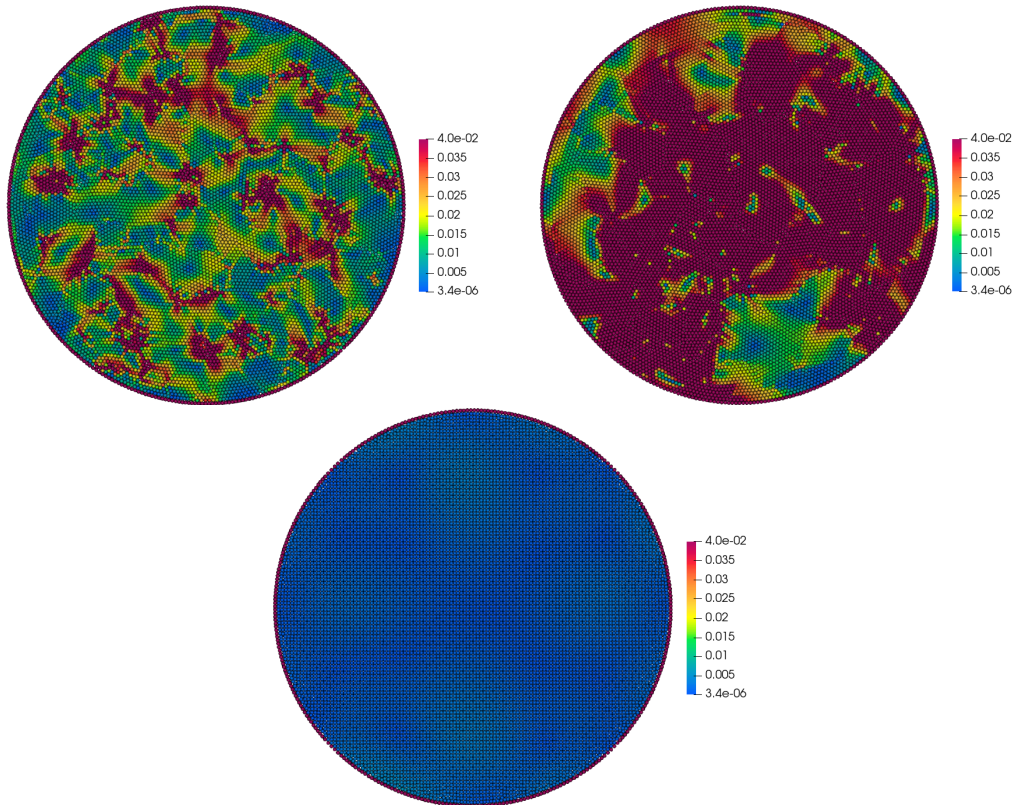


FIGURE 2.9: Particle relaxation residue using the Wendland (top left panel), Laguerre-Wendland (top right panel) and truncated Laguerre-Gauss (bottom panel) kernels at the final relaxation iteration with the spatial resolution $dp = 1/60$.

TABLE 2.4: Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the truncated Laguerre-Gauss kernel with different κ without the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Laguerre-Gauss	1.1	0.021	0.259	0.614	1.253
	1.3	0.004	0.123	0.011	0.204
	1.6	0.007	0.210	0.822	0.817

TABLE 2.5: Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the truncated Laguerre-Gauss kernel with different κ with the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Laguerre-Gauss	1.1	0.021	0.061	0.603	0.448
	1.3	0.004	0.004	0.011	0.009
	1.6	0.006	0.024	0.784	0.364

the original truncated Gaussian kernel, might exhibit pair-instability at $\kappa = 1.3$, the results indicate that such instability does not occur. Furthermore, employing the truncated Laguerre-Gauss kernel ensures improved numerical stability while obtaining an optimal particle distribution.

2.9.4 Performance in accuracy

In this section, we evaluate the SPH approximation for the derivatives of the mentioned functions with the truncated Laguerre-Gauss kernel, which aims to further confirm the kernel's accuracy. Specifically, the truncation error for the derivative of the function $f = 1$ matches the relaxation residue presented in Eq. (2.45).

Tables 2.4 and 2.5 display the truncation errors for the truncated Laguerre-Gauss kernel. The results indicate that at $\kappa = 1.1$, the truncation error is relatively large due to the insufficient neighbouring particles. Conversely, at $\kappa = 1.6$, large errors are attributed to the pair-instability issue caused by an excessive number of neighbouring particles. To avoid excessive computational costs, $\kappa = 1.3$ is typically applied in practice. For $\kappa = 1.3$, the truncation errors of the truncated Laguerre-Gauss kernel in Table 2.4 and of the Wendland kernel in Table 2.2 are comparable when the kernel correction matrix is not applied. However, the proposed kernel with the kernel correction matrix shown in Table 2.5 significantly reduces the truncation error compared to both the Wendland and Laguerre-Wendland kernels shown in Table 2.3, indicating its considerably improved accuracy. Furthermore, the truncated Laguerre-Gauss kernel outperforms the other two kernels regarding average relaxation residue and the range of maximum and minimum residue variations.

2.9.5 Convergence analysis

In this section, we conduct a convergence analysis using the circular geometry. The scalar field is initially defined by the following function:

$$f(x) = e^{-x^2/0.1}. \quad (2.69)$$

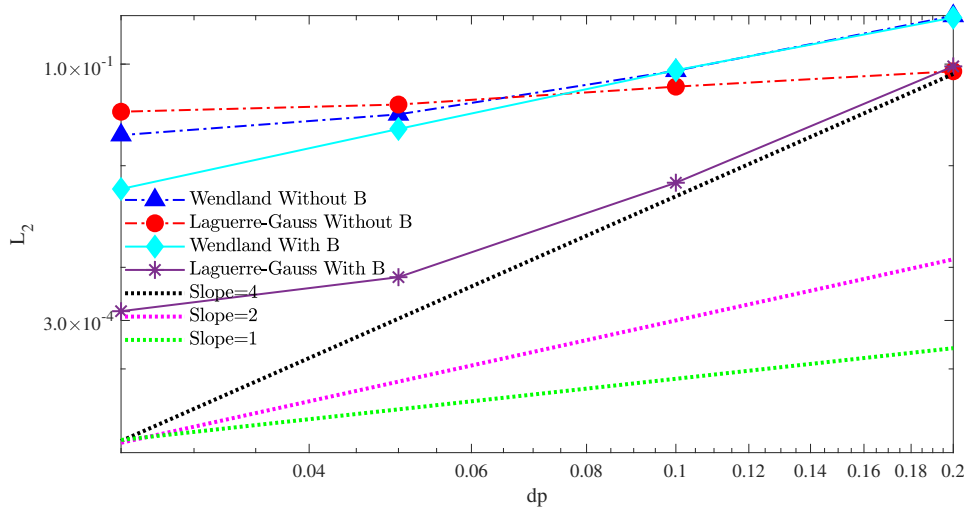


FIGURE 2.10: Convergence study using the Wendland and Laguerre-Gauss kernel with different resolutions by comparing the different theoretical order curves.

Following Ref. [71], the L_2 -norm error for the estimated gradient operators can be described by

$$L_2(df/dx) = \frac{\sqrt{\sum_i |(\nabla f_i)^{Theoretical} - (\nabla f_i)^{Numerical}|^2}}{\sqrt{\sum_i |(\nabla f_i)^{Theoretical}|^2}}, \quad (2.70)$$

where $(\nabla f_i)^{Numerical}$ is the strong form as Eq. (2.13).

Figure 2.10 presents the convergence analysis involving the Wendland and truncated Laguerre-Gauss kernels across different particle resolutions with a constant parameter of $\kappa = 1.3$. Without the kernel correction matrix, as resolution increases, the Wendland kernel initially converges at a second-order rate and gradually degenerates to saturation, where the integration error dominates. In contrast, the truncated Laguerre-Gauss kernel exhibits significantly greater accuracy at lower resolutions compared to the Wendland kernel, however, its accuracy degenerates to saturation quickly with increasing resolution. This behaviour suggests that the truncated Laguerre-Gauss kernel obtains higher accuracy at low resolution due to the smaller smoothing error, and its integration error is still larger than that of the Wendland kernel.

Using the kernel correction matrix, the Wendland kernel consistently maintains a second-order convergence rate. Conversely, the truncated Laguerre-Gauss kernel demonstrates fourth-order convergence at lower resolutions but degenerates to second-order as resolution increases. This behaviour aligns with expectations: for the Wendland kernel, both smoothing and integration errors result in second-order convergence. In the case of the truncated Laguerre-Gauss kernel, the fourth-order smoothing error dominates at low resolutions, whereas at higher resolutions, the second-order integration error becomes dominant.

The convergence study highlights that the truncated Laguerre-Gauss kernel consistently produces significantly lower errors compared to the Wendland kernel, regardless of particle resolution, when using the kernel correction matrix. Note that

such improvement is achieved with the same compact support and kernel correction matrix as the Wendland kernel. Additionally, at high resolutions, the truncated Laguerre-Gauss kernel reaches equivalent accuracy with a small fraction of particles (approximately $1/16$ in two dimensions, as indicated in Figure 2.10).

2.10 Summary

This chapter presents the fundamental and theoretical concepts of the SPH method, integral interpolation and particle approximation. Compressible and weakly compressible SPH methods, encompassing the equation of state, Riemann-based SPH discretizations of conservation equations, and time integration, are discussed. Additionally, a brief overview of the Eulerian extensions is provided, covering dissipation limiters, a kernel correction matrix, and a particle relaxation technique. A rigorous comparison between the Eulerian SPH and FVM, as well as the realization of FVM within the SPH codebase, are also investigated. To enhance equivalence between these two methods, a detailed discussion of the FVM-equivalent boundary algorithm is included. Furthermore, numerical accuracy enhancements, including WENO reconstruction and the fourth-order Laguerre-Gauss kernel, are elaborated upon.

Chapter 3

Summaries of Publication

In this chapter, the relevant publications of this thesis are briefly summarized.

3.1 FVM realization in Eulerian SPH: A comparative study within unified codebase

Zhentong Wang, Oskar J. Haidn, Nikolaus A. Adams, Xiangyu Hu

3.1.1 Summary of the publication

In the CFD community, the Eulerian SPH method based on kernel function can be considered as a mesh-based FVM [36, 26, 37]. The analog has been detailed by Neuhauser [38] where a FVM discretization can leverage the Riemann solver for the ALE SPH method within the same software. However, to which level can such analog reach between the Eulerian SPH and FVM has not been explored yet.

In this paper, we first introduce a form of the Eulerian SPH method where the kernel gradient matrix [40] given by

$$(\mathbf{e}_{ij}^{SPH})' = \frac{\nabla' W_{ij}}{|\nabla' W_{ij}|} \quad (3.1)$$

is introduced to release the constraints associated with the normal direction of interfaces. Once the constraint is released, a virtual mesh structure emerges, which can be regarded as being generated by the SPH method, with $(\mathbf{e}_{ij}^{SPH})'$ orthogonal to the corresponding virtual interface, analogous to the traditional mesh produced by mesh-generation software. Consequently, $(\mathbf{e}_{ij}^{SPH})'$ is analogous to $\mathbf{n}_{ij}^{FVM}$ in traditional mesh-based FVM. Additionally, the second task is to implement the traditional FVM formulation directly within the open-source SPH library, SPHinXsys [41]. This is achieved by developing a parser to extract essential data from the ".msh" mesh file generated by ICEM. A rigorous comparison can finally be conducted by using a shared solver for both methods within a unified SPH codebase. As shown in Figure 3.1, the Eulerian SPH method has the obvious advantage of obtaining much smoother outcomes as its kernel's isotropic property compared with the mesh-based FVM. Moreover, a faster convergence rate is achieved by Eulerian SPH method using the fourth-order Laguerre-Gauss kernel compared with FVM, as demonstrated as Figure 3.2. Furthermore, the computational efficiency listed in Table 3.1 has been conducted, indicating that the computational efficiency using the Eulerian SPH method is lower than that in FVM.

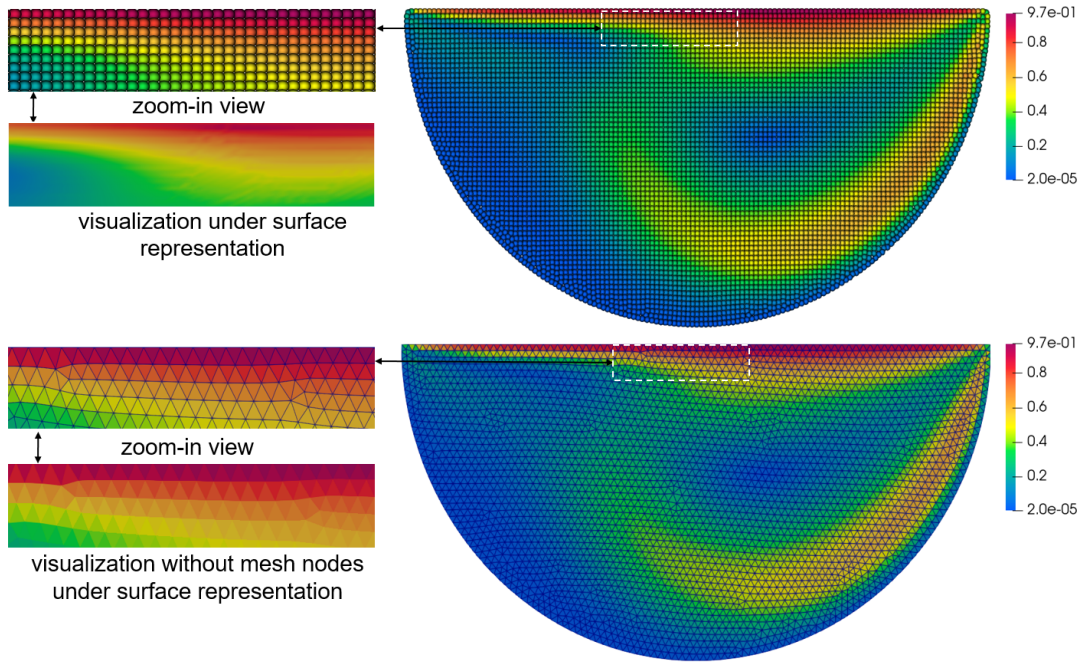


FIGURE 3.1: Lid-driven semi-circular cavity flow: The velocity contour and its zoom-in view obtained by ESPH (top panel), and FVM (bottom panel).

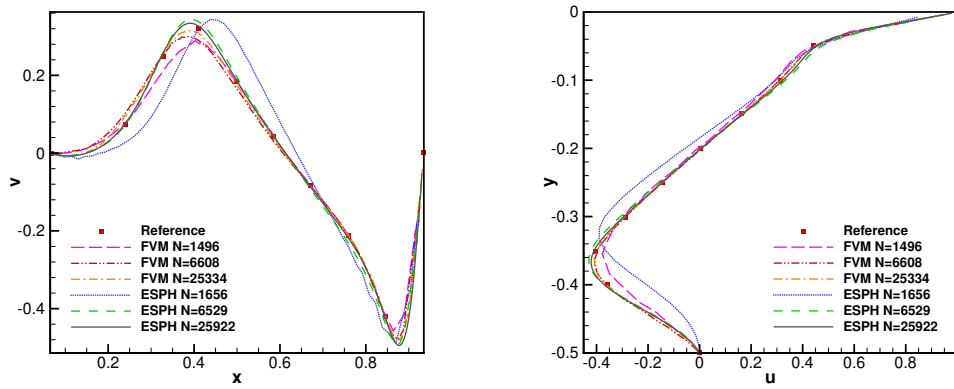


FIGURE 3.2: Lid-driven semi-circular cavity flow: The horizontal velocity component u along $x = 0.5L$ (left panel) and the vertical velocity component v along $y = -0.25L$ (right panel) obtained by ESPH and FVM.

TABLE 3.1: Lid-driven semi-circular cavity flow with $Re = 1000$: Computational efficiency comparison between FVM and ESPH for the physical time $t = 30$.

	Element/Particle number	CPU wall-clock time	Iteration number	α
FVM	6608	68.99s	197677	0.394
ESPH	6529	174.78s	108768	

3.1.2 Individual novelty of the author

The article [39] was published in the internationally peer-reviewed *Computer Physics Communications*. My contribution included developing the method and the corresponding implementation code. I conducted the simulations, performed result analysis, and drafted the manuscript for the publication.

3.2 An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework

Zhentong Wang, Oskar J. Haidn, Xiangyu Hu

3.2.1 Summary of the publication

The integration of mesh-based methods like the FVM with meshless approaches such as SPH plays a vital role in addressing complex fluid dynamics challenges. However, it is acknowledged that weak coupling between these different techniques presents inherent challenges. Wang et al. [39] utilized a unified solver and codebase to compare accuracy and efficiency for FVM and SPH methods, yet the handling of boundary conditions remains distinct between the particle-based and mesh-based frameworks. Consequently, incorporating the FVM boundary algorithm into the Eulerian SPH method is essential for enhancing the algorithmic equivalency and proving the foundation for establishing the strong coupling between these two methods.

In this study, we analyze the FVM boundary algorithm, which ensures zero-order consistency for cells in the boundaries, and propose a new FVM-equivalent boundary algorithm into the Eulerian SPH method. During the implementation of the boundary algorithm within the SPH codebase, additional storage space is allocated for particles in the boundaries to retain necessary information, thereby fulfilling the FVM's boundary requirements. In the original boundary approach [65, 39, 44] within the SPH framework, boundary conditions, such as the wall boundary condition [65], are typically applied to predefined particles within the contact configurations, in addition to the fluid particles within the inner configuration. In contrast, the present algorithm requires only the inner configuration. As shown in Figure 3.3, numerical simulation of the classic benchmark case of Poiseuille flow shows both method's velocity and error analysis, validating that the new boundary algorithm is accurate and exhibits an enhanced convergence rate.

3.2.2 Individual novelty of the author

The article [42] was published in the internationally peer-reviewed *Computer Physics Communications*. My contribution included developing the method and the corresponding implementation code. I conducted the simulations, performed result analysis, and drafted the manuscript for the publication.

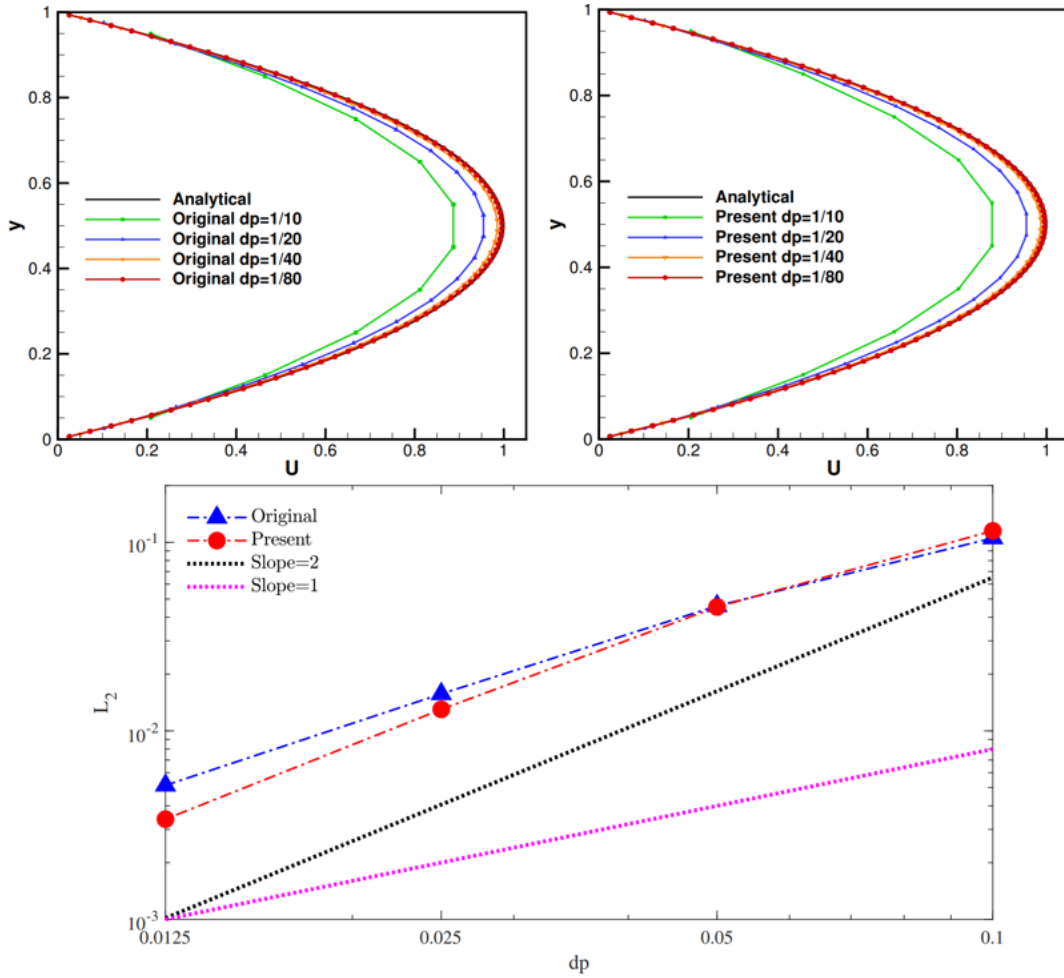


FIGURE 3.3: Poiseuille flow's velocity profile using original and new boundary algorithms, L_2 error and convergence analysis.

3.3 An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows

Zhentong Wang, Chi Zhang, Oskar J. Haidn, Xiangyu Hu

3.3.1 Summary of the publication

To improve the numerical accuracy in the Riemann-based SPH method, the high-order reconstruction was introduced to decrease the integration error. High-order reconstruction, i.e., weighted essentially non-oscillatory (WENO), used in the Eulerian Godunov method, can achieve higher accuracy and less numerical dissipation [72]. A one-dimensional fifth-order WENO reconstruction extending to simulate the multi-dimensional problem in the WCSPH framework is developed with its simple and computational efficiency [43]. However, this method is only applied to the Lagrangian WCSPH, making its study in the Eulerian SPH for compressible and incompressible flows imperfect and different effects of the WENO reconstruction in the smooth region and contact discontinuity region lacking.

In this paper, we first exploit low-dissipation linearised and HLLC Riemann solvers for incompressible and compressible flows. Then, a kernel gradient correction is introduced to achieve first-order consistency and a one-dimensional WENO reconstruction is adopted to improve the overall numerical accuracy further and the shock-capturing performance in the contact discontinuity. The different effects of the WENO reconstruction on smooth region and contact discontinuity are also explored. Figure 3.4 shows the density profile using the WENO compared with a high-resolution scheme [73], showing the ESPH-WENO exhibits a much sharper density profile and less dissipation, implying that the WENO reconstruction effectively improves the numerical accuracy, especially at contact regions where the limiter has little effect.

3.3.2 Individual novelty of the author

The article [44] was published in the internationally peer-reviewed *Journal of Hydrodynamics*. My contribution included developing the method and the corresponding implementation code. I conducted the simulations, performed result analysis, and drafted the manuscript for the publication.

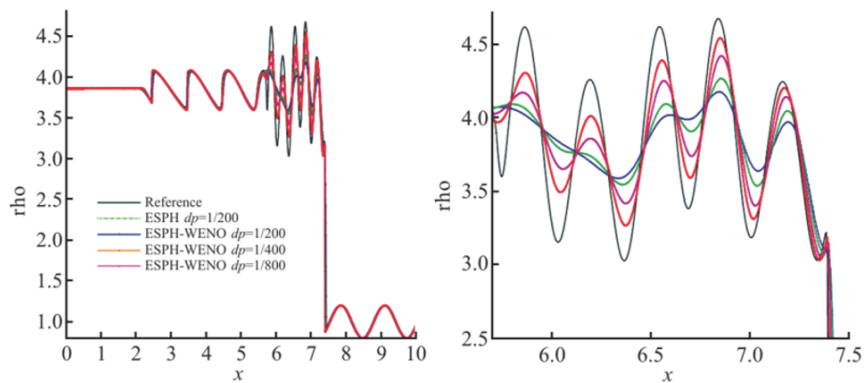


FIGURE 3.4: Shu-Osher problem: Density profile (left panel) and the zoom-in view (right panel).

3.4 A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids

Zhentong Wang, Bo Zhang, Oskar J. Haidn, Xiangyu Hu

3.4.1 Summary of the publication

Generally, two factors determine the accuracy of the SPH method, i.e. smoothing and integration errors [24, 26]. To achieve higher accuracy of SPH approximation, one can resort to the high-order kernel methods [45, 74] using kernels characterized with non-monotonic, non-positive regions and high-order vanishing moments for less smoothing error and sufficiently large compact support for less integration error. However, to avoid the excessive computational effort for extensive compact supports, the current mainstream kernel functions used in SPH, such as B-spline, Wendland, truncated Gaussian (for compact support) and others, are monotonic and give 2nd-order smoothing error as only the first moment vanishes. Litvinov et al. [26] found that, under relaxed particle distribution, the convergence rate of integration error (8th-order respected to support radius) can be much higher than that of smoothing error. Therefore, two non-monotonic, namely truncated Laguerre–Gauss and Laguerre–Wendland, kernels with 4th-order smoothing error, have been considered for improving the overall accuracy of SPH approximations. However, these kernels have not been applied for practical SPH algorithms because, for moderate compact support, the actual integration error of the Laguerre–Wendland kernel is much larger than that obtained by the original 2nd-order Wendland kernel. Note that the truncated Laguerre–Gauss kernel has not been tested in Ref. [26], probably due to the pair-instability problem of the original truncated Gaussian kernel.

In this work, the extensive integration error of the Laguerre–Wendland kernel in moderate compact support is revisited, and the source of the error has been analyzed. Furthermore, the truncation error of the truncated Laguerre–Gauss kernel is investigated and is chosen to improve the overall accuracy of SPH approximations. The truncated Laguerre–Gauss kernel exhibits fourth-order convergence at low resolution but degenerates to second-order with increasing resolutions. Besides, independent of particle resolution, with the kernel correction matrix, the truncated Laguerre–Gauss kernel always introduces much less errors than the Wendland kernel. In the benchmark case of flow around a cylinder, Figure 3.5 shows the velocity contour ranging from 1.8×10^{-3} to 1.5 using the Wendland and Laguerre–Gauss kernels with the Reynolds number $Re = 100$ under the resolution $dp = 1/6$ at $t = 300$. It can be seen that the method using the Laguerre–Gauss kernel efficiently captures the vortex street while that using the Wendland kernel fails to present the same physical phenomenon, indicating that the proposed kernel is more effective in reducing numerical errors and thus obtaining more accurate results. The computations were conducted on an Intel Core i7-10700 2.90 GHz 8-core desktop computer. The CPU wall-clock times for the Eulerian SPH method, using the Wendland and Laguerre–Gauss kernels with a spatial resolution of $dp = 1/10$, were 4604.81s and 4818.91s, respectively. This indicates that the computational efficiency of both kernels is comparable.

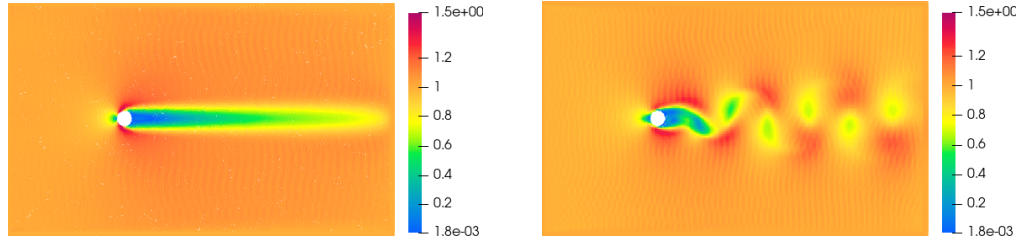


FIGURE 3.5: Flow around a cylinder: Velocity contour using the Wendland (left panel) and Laguerre-Gauss (right panel) kernels with $Re = 100$.

3.4.2 Individual novelty of the author

The article [32] was published in the internationally peer-reviewed *Journal of Computational Physics*. My contribution included developing the method and the corresponding implementation code. I conducted the simulations, performed result analysis, and drafted the manuscript for the publication.

Chapter 4

Conclusions and outlooks

Four primary motivations drive this thesis: to achieve a rigorous comparison by implementing the FVM within the Eulerian SPH codebase; to enhance the consistency between the FVM and Eulerian SPH methods through the development of the FVM-equivalent boundary in the Eulerian SPH method; to improve the accuracy by incorporating the WENO reconstruction scheme into the Eulerian SPH; and to enhance accuracy by employing the fourth-order Laguerre-Gauss kernel in the Eulerian SPH. These objectives have been successfully accomplished across multiple applications. Here, I summarize this thesis and provide discussions and future research remarks.

4.1 Conclusions

To explore the similarities and differences between FVM and Eulerian SPH methods, Neuhauser [38] provides an in-depth analysis of their analogies, demonstrating that FVM discretization can effectively reuse the Riemann solver developed for the ALE SPH method within the same software package. However, the rigorous comparison remains unachievable due to the limitation in the Eulerian SPH method: the flux direction must align with the positional direction.

In Paper I, the kernel gradient correction matrix is applied to release the constraint in the Eulerian SPH method, facilitating the adjustment of flux properties. Then, the Eulerian SPH method can achieve algorithmic equivalence to the FVM, particularly in zero-order consistency, interface flux direction, and flux weighting from neighbouring particles. Additionally, the mesh-based FVM is realized in the same SPH codebase using a shared solver, encompassing consistent discretized conservation equations, flux data structures, Riemann solvers, dissipation limiters, and time integration schemes. These approaches enable a rigorous comparison of computational efficiency and performance between these two methods. Tests on compressible and incompressible flows validate that while the mesh-based FVM achieves higher computational efficiency due to fewer neighbours, Eulerian SPH excels in producing smoother results due to the kernel's isotropic characteristics and a faster convergence rate when employing the high-order kernel. Note that computational time and iteration count in mesh-based FVM are influenced by mesh quality, as the minimum distance between grid nodes limits the time step size.

Although the FVM has been incorporated into the SPH codebase and allows for a comparison between the two methods, the boundary implementation differs. To enhance the equivalence between these methods, a consistent boundary algorithm is required. In Paper II, we introduce a particle relaxation technique to generate a body-fitted particle distribution where particles not in the boundaries meet zero-order consistency. By analyzing FVM's boundary algorithm, we incorporate a FVM-equivalent approach within the Eulerian SPH framework, ensuring that particles

in the boundaries achieve strict zero-order consistency. Furthermore, several numerical tests, including both compressible and incompressible flows, validate the stability, accuracy, and improved convergence rate of the proposed boundary algorithm compared to the original. An additional benefit is enhanced computational efficiency.

While the Lagrangian SPH method provides topological flexibility, which results in poor particle distribution, the Eulerian SPH method mitigates this issue by keeping particles fixed, preserving the optimal initial distribution throughout the simulation [28, 29]. Enhancing numerical accuracy can be effectively achieved through high-order techniques. While WENO has been applied within the Lagrangian framework, its use in the Eulerian method remains unexplored. In Paper III, we present a WENO reconstruction to enhance the accuracy of the ESPH method. This approach involves constructing a high-accuracy WENO reconstruction of the Riemann states, achieved by introducing a one-dimensional 4-point stencil along interacting particle pairs. Several numerical examples of compressible and incompressible flows are analyzed and have validated the proposed method's improved accuracy, robustness, and versatility.

Although WENO reconstruction enhances numerical accuracy, it significantly increases computational costs. Alternatively, an effective and simple method to improve the accuracy of SPH approximations is employing high-order kernel methods [45]. These kernels feature non-monotonic and non-positive regions with high-order vanishing moments, reducing smoothing error, and have sufficiently large compact support to minimize integration error. However, such high-order kernels also lead to increased computational demands due to their extended support. Among these, a fourth-order Laguerre-Gauss kernel remains untested. In Paper IV, the excessive integration error of the Laguerre-Wendland kernel is revisited, and the analysis of the truncated Laguerre-Gauss kernel revealed that it introduces substantially less truncation error and relaxation residue compared to the Wendland and Laguerre-Wendland kernels, resulting in notable improvements in stability and accuracy. Furthermore, several 2D and 3D numerical tests for fluid dynamics have validated that the proposed kernel achieves significantly higher accuracy than the Wendland kernel, with comparable computational efficiency.

4.2 Outlook

The present work can be further investigated in the following aspects:

- This study conducts a rigorous comparison between the FVM and Eulerian SPH methods with second-order accuracy. Expanding this comparison using high-order schemes is necessary for a more in-depth evaluation of their performance.
- The current codebase supports both mesh-based FVM and meshless SPH methods. Developing a strong coupling between mesh-based and mesh-free methods using a shared solver within a unified framework is significantly beneficial as it improves stability and computational efficiency.
- The fourth-order Laguerre-Gauss kernel has demonstrated high numerical accuracy without compromising computational efficiency, though its use is restricted to the Eulerian SPH framework. Extending its application to the Lagrangian framework is necessary and essential.
- Given the promising numerical performance of the Laguerre-Gauss kernel, it becomes crucial to explore whether other high-order kernels demonstrate comparable or improved characteristics.

Bibliography

- [1] Faraz Afshari et al. "On numerical methods; optimization of CFD solution to evaluate fluid flow around a sample object at low Re numbers". In: *Mathematics and Computers in Simulation* 152 (2018), pp. 51–68.
- [2] Joseph O'connor et al. "Eulerian incompressible smoothed particle hydrodynamics on multiple GPUs". In: *Computer Physics Communications* 273 (2022), p. 108263.
- [3] John David Anderson and John Wendt. *Computational fluid dynamics*. Vol. 206. Springer, 1995.
- [4] Olgierd Cecil Zienkiewicz and Robert Leroy Taylor. *The finite element method: solid mechanics*. Vol. 2. Butterworth-heinemann, 2000.
- [5] Robert Eymard, Thierry Gallouët, and Raphaële Herbin. "Finite volume methods". In: *Handbook of numerical analysis* 7 (2000), pp. 713–1018.
- [6] Leon B Lucy. "A numerical approach to the testing of the fission hypothesis". In: *Astronomical Journal*, vol. 82, Dec. 1977, p. 1013-1024. 82 (1977), pp. 1013–1024.
- [7] Joe J Monaghan. "SPH and Riemann solvers". In: *Journal of Computational Physics* 136.2 (1997), pp. 298–307.
- [8] Seiichi Koshizuka and Yoshiaki Oka. "Moving-particle semi-implicit method for fragmentation of incompressible fluid". In: *Nuclear science and engineering* 123.3 (1996), pp. 421–434.
- [9] Antonio A Munjiza. *The combined finite-discrete element method*. John Wiley & Sons, 2004.
- [10] Xiang Yu Hu and Nikolaus A Adams. "A multi-phase SPH method for macroscopic and mesoscopic flows". In: *Journal of Computational Physics* 213.2 (2006), pp. 844–861.
- [11] Robert A Gingold and Joseph J Monaghan. "Smoothed particle hydrodynamics: theory and application to non-spherical stars". In: *Monthly notices of the royal astronomical society* 181.3 (1977), pp. 375–389.
- [12] L. D. Libersky and A. G. Petschek. "Smooth Particle Hydrodynamics with strength of materials". In: *Advances in the free-Lagrange method including contributions on adaptive gridding and the smooth particle hydrodynamics method*. Springer, 1991, pp. 248–257.
- [13] Chi Zhang et al. "Review on Smoothed Particle Hydrodynamics: Methodology development and recent achievement". In: *arXiv preprint arXiv:2205.03074* (2022).
- [14] Min Luo, Abbas Khayyer, and Pengzhi Lin. "Particle methods in ocean and coastal engineering". In: *Applied Ocean Research* 114 (2021), p. 102734.

- [15] Hitoshi Gotoh, Abbas Khayyer, and Yuma Shimizu. "Entirely Lagrangian meshfree computational methods for hydroelastic fluid-structure interactions in ocean engineering—Reliability, adaptivity and generality". In: *Applied Ocean Research* 115 (2021), p. 102822.
- [16] Joe J Monaghan. "Smoothed particle hydrodynamics". In: *Reports on progress in physics* 68.8 (2005), p. 1703.
- [17] MB Liu and GR2593940 Liu. "Smoothed particle hydrodynamics (SPH): an overview and recent developments". In: *Archives of computational methods in engineering* 17 (2010), pp. 25–76.
- [18] Joseph J Monaghan. "Smoothed particle hydrodynamics and its diverse applications". In: *Annual Review of Fluid Mechanics* 44.1 (2012), pp. 323–346.
- [19] Lei Li et al. "Meshfree simulation and experimental validation of extreme thermomechanical conditions in friction stir extrusion". In: *Computational Particle Mechanics* (2021), pp. 1–21.
- [20] Moncho Gomez-Gesteira et al. "State-of-the-art of classical SPH for free-surface flows". In: *Journal of Hydraulic Research* 48.sup1 (2010), pp. 6–27.
- [21] Massoud Rezavand, Chi Zhang, and Xiangyu Hu. "A weakly compressible SPH method for violent multi-phase flows with high density ratio". In: *Journal of Computational Physics* 402 (2020), p. 109092.
- [22] C. Antoci, M. Gallati, and S. Sibilla. "Numerical simulation of fluid–structure interaction by SPH". In: *Computers & Structures* 85.11 (2007), pp. 879–890.
- [23] Chi Zhang, Massoud Rezavand, and Xiangyu Hu. "A multi-resolution SPH method for fluid-structure interactions". In: *Journal of Computational Physics* 429 (2021), p. 110028.
- [24] Nathan J Quinlan, Mihai Basa, and Martin Lastiwka. "Truncation error in mesh-free particle methods". In: *International Journal for Numerical Methods in Engineering* 66.13 (2006), pp. 2064–2085.
- [25] Stefan Adami, XY Hu, and Nikolaus A Adams. "A transport-velocity formulation for smoothed particle hydrodynamics". In: *Journal of Computational Physics* 241 (2013), pp. 292–307.
- [26] Sergey Litvinov, XY Hu, and Nikolaus A Adams. "Towards consistence and convergence of conservative SPH approximations". In: *Journal of Computational Physics* 301 (2015), pp. 394–401.
- [27] Renato Vacondio et al. "Grand challenges for Smoothed Particle Hydrodynamics numerical schemes". In: *Computational Particle Mechanics* 8.3 (2021), pp. 575–588.
- [28] Mihai Basa, Nathan J Quinlan, and Martin Lastiwka. "Robustness and accuracy of SPH formulations for viscous flow". In: *International Journal for Numerical Methods in Fluids* 60.10 (2009), pp. 1127–1148.
- [29] AMA Nasar et al. "Eulerian weakly compressible smoothed particle hydrodynamics (SPH) with the immersed boundary method for thin slender bodies". In: *Journal of Fluids and Structures* 84 (2019), pp. 263–282.
- [30] Rodrigue Keou Noutcheuwa and Robert G Owens. "A new incompressible smoothed particle hydrodynamics-immersed boundary method". In: *Int. J. Numer. Anal. Model. Series B* 3.2 (2012), pp. 126–167.

- [31] SJ Lind and PK Stansby. "Investigations into higher-order incompressible SPH". In: *Proc. 10th SPHERIC Int. Workshop*. 2015, pp. 131–138.
- [32] Zhentong Wang et al. "A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids". In: *Journal of Computational Physics* (2024), p. 113385.
- [33] Abouzied Mohamed Nasar. *Eulerian and Lagrangian smoothed particle hydrodynamics as models for the interaction of fluids and flexible structures in biomedical flows*. The University of Manchester (United Kingdom), 2016.
- [34] Chi Zhang, Massoud Rezavand, and Xiangyu Hu. "Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics". In: *Journal of Computational Physics* 404 (2020), p. 109135.
- [35] Zhentong Wang, Oskar J Haidn, and Xiangyu Hu. "Incorporating the algorithm for the boundary condition from FVM into the framework of Eulerian SPH". In: *arXiv preprint arXiv:2312.03542* (2023).
- [36] Jean Paul Vila. "On particle weighted methods and smooth particle hydrodynamics". In: *Mathematical models and methods in applied sciences* 9.02 (1999), pp. 161–209.
- [37] Chi Zhang, XY Hu, and Nikolaus A Adams. "A weakly compressible SPH method based on a low-dissipation Riemann solver". In: *Journal of Computational Physics* 335 (2017), pp. 605–620.
- [38] Magdalena Neuhauser. "Development of a coupled SPH-ALE/Finite Volume method for the simulation of transient flows in hydraulic machines". PhD thesis. Ecole Centrale de Lyon, 2014.
- [39] Zhentong Wang et al. "FVM realization in Eulerian SPH: A comparative study within unified codebase". In: *Computer Physics Communications* (2025), p. 109683.
- [40] P. W. Randles and L. D. Libersky. "Smoothed particle hydrodynamics: some recent improvements and applications". In: *Computer Methods in Applied Mechanics and Engineering* 139.1 (1996), pp. 375–408.
- [41] Chi Zhang et al. "SPHinXsys: An open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics". In: *Computer Physics Communications* 267 (2021), p. 108066.
- [42] Zhentong Wang, Oskar J Haidn, and Xiangyu Hu. "An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework". In: *Computer Physics Communications* 307 (2025), p. 109429.
- [43] Chi Zhang et al. "A weakly compressible SPH method with WENO reconstruction". In: *Journal of Computational Physics* 392 (2019), pp. 1–18.
- [44] Zhentong Wang et al. "An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows". In: *Journal of Hydrodynamics* 35.2 (2023), pp. 210–221.
- [45] Steven J Lind and PK3554142 Stansby. "High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion". In: *Journal of Computational Physics* 326 (2016), pp. 290–311.
- [46] Chi Zhang et al. "Smoothed particle hydrodynamics: Methodology development and recent achievement". In: *Journal of Hydrodynamics* 34.5 (2022), pp. 767–805.

- [47] Holger Wendland. "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree". In: *Advances in computational Mathematics* 4 (1995), pp. 389–396.
- [48] Joe J Monaghan. "Smoothed particle hydrodynamics". In: *Annual review of astronomy and astrophysics*. Vol. 30 (A93-25826 09-90), p. 543-574. 30 (1992), pp. 543–574.
- [49] Gui-Rong Liu and Moubin B Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World scientific, 2003.
- [50] MB Liu, GR Liu, and KY Lam. "Constructing smoothing functions in smoothed particle hydrodynamics with applications". In: *Journal of Computational and applied Mathematics* 155.2 (2003), pp. 263–284.
- [51] Joseph J Monaghan and John C Lattanzio. "A refined particle method for astrophysical problems". In: *Astronomy and Astrophysics (ISSN 0004-6361)*, vol. 149, no. 1, Aug. 1985, p. 135-143. 149 (1985), pp. 135–143.
- [52] Daniel J Price. "Magnetic fields in Astrophysics". In: *Ph. D. Thesis* (2004).
- [53] J. W. Swegle, D. L. Hicks, and S. W. Attaway. "Smoothed Particle Hydrodynamics stability analysis". In: *Journal of computational physics* 116.1 (1995), pp. 123–134.
- [54] Ha H Bui and Giang D Nguyen. "Smoothed particle hydrodynamics (SPH) and its applications in geomechanics: From solid fracture to granular behaviour and multiphase flows in porous media". In: *Computers and Geotechnics* 138 (2021), p. 104315.
- [55] Eleuterio F Toro, Michael Spruce, and William Speares. "Restoration of the contact surface in the HLL-Riemann solver". In: *Shock waves* 4 (1994), pp. 25–34.
- [56] Eleuterio F Toro. "The hllc riemann solver". In: *Shock waves* 29.8 (2019), pp. 1065–1082.
- [57] Amiram Harten, Peter D Lax, and Bram van Leer. "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws". In: *SIAM review* 25.1 (1983), pp. 35–61.
- [58] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.
- [59] Zhentong Wang et al. "Extended Eulerian SPH and its realization of FVM". In: *arXiv preprint arXiv:2309.01596* (2023).
- [60] Zhentong Wang, Oskar J Haidn, and Xiangyu Hu. "Efficient implementation of transport velocity formulation". In: *Journal of Computational Physics* (2025), p. 114203.
- [61] Bram Van Leer. "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method". In: *Journal of computational Physics* 32.1 (1979), pp. 101–136.
- [62] Phillip Colella and Paul R Woodward. "The piecewise parabolic method (PPM) for gas-dynamical simulations". In: *Journal of computational physics* 54.1 (1984), pp. 174–201.
- [63] Vito Zago et al. "Overcoming excessive numerical dissipation in SPH modeling of water waves". In: *Coastal Engineering* 170 (2021), p. 104018.

- [64] Yujie Zhu et al. "A CAD-compatible body-fitted particle generator for arbitrarily complex geometry and its application to wave-structure interaction". In: *Journal of Hydrodynamics* 33.2 (2021), pp. 195–206.
- [65] Stefan Adami, Xiangyu Y Hu, and Nikolaus A Adams. "A generalized wall boundary condition for smoothed particle hydrodynamics". In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075.
- [66] Joe J Monaghan and Jules B Kajtar. "SPH particle boundary forces for arbitrary boundaries". In: *Computer physics communications* 180.10 (2009), pp. 1811–1820.
- [67] Martin Ferrand et al. "Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method". In: *International Journal for Numerical Methods in Fluids* 71.4 (2013), pp. 446–472.
- [68] François Dubois. "3.1 partial Riemann problem, boundary conditions, and gas dynamics". In: *Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to Large Scale Computers* (2001), p. 16.
- [69] Lin Fu, Xiangyu Y Hu, and Nikolaus A Adams. "A family of high-order targeted ENO schemes for compressible-fluid simulations". In: *Journal of Computational Physics* 305 (2016), pp. 333–359.
- [70] VK Triteschler et al. "Numerical simulation of a Richtmyer–Meshkov instability with an adaptive central-upwind sixth-order WENO scheme". In: *Physica Scripta* 2013.T155 (2013), p. 014016.
- [71] Alban Vergnaud, Guillaume Oger, and David Le Touzé. "Investigations on a high order SPH scheme using WENO reconstruction". In: *Journal of Computational Physics* 477 (2023), p. 111889.
- [72] Sergio Pirozzoli. "Numerical methods for high-speed flows". In: *Annual review of fluid mechanics* 43.1 (2011), pp. 163–194.
- [73] Yujie Zhu and Xiangyu Hu. "An L2-norm regularized incremental-stencil WENO scheme for compressible flows". In: *Computers & Fluids* 213 (2020), p. 104721.
- [74] Georgios Fourtakas et al. "An Eulerian–Lagrangian incompressible SPH formulation (ELI-SPH) connected with a sharp interface". In: *Computer Methods in Applied Mechanics and Engineering* 329 (2018), pp. 532–552.

Appendix A

Original journal papers

Here, the peer-reviewed journal publications of the present work are attached.

A.1 Paper I

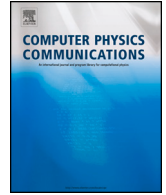
Zhentong Wang, Oskar J. Haidn, Nikolaus A. Adams, Xiangyu Hu

FVM realization in Eulerian SPH: A comparative study within unified codebase

In *Computer Physics Communications*, Volume 314, 2025, p. 109683, DOI: <https://doi.org/10.1016/j.cpc.2025.109683>.

Copyright © 2025 Elsevier. Reprinted with permission.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.



Computational Physics

FVM realization in Eulerian SPH: A comparative study within unified codebase

Zhentong Wang, Chi Zhang, Oskar J. Haidn, Nikolaus A. Adams, Xiangyu Hu ^{ID,*}

TUM School of Engineering and Design, Technical University of Munich, Garching, 85747, Germany

ARTICLE INFO

The review of this paper was arranged by Prof. Peter Vincent

Keywords:

Eulerian smoothed particle hydrodynamics
Finite volume method
Rigorous comparison
Kernel gradient correction
Algorithmic equivalency

ABSTRACT

Eulerian smoothed particle hydrodynamics (Eulerian SPH) is considered a potential meshless alternative to a traditional Eulerian mesh-based finite volume method (FVM) in computational fluid dynamics (CFD). While researchers have analyzed the differences between these two methods, a rigorous comparison of their performance and computational efficiency is hindered by the following two challenges: Firstly, the Eulerian SPH framework faces a constraint related to the normal direction of interfaces in pairwise particle interactions, which prevents achieving an equivalent algorithm of FVM; Secondly, there is no unified solver available that can be applied to both Eulerian SPH and FVM methods. To address the former constraint, this paper implements a certain form of Eulerian SPH method, where a kernel gradient correction is introduced to release the constraint. To address the latter constraint, the paper realizes the mesh-based FVM within an open-source SPH library using a shared solver, i.e. single algorithm for two methods, through developing a parser that extracts necessary information from the “.msh” format file exported from a commercial pre-processing tool, ICEM (Integrated Computer Engineering and Manufacturing). Several 2D and 3D numerical simulations using both methods within a unified codebase demonstrate that compared to the mesh-based FVM, the Eulerian SPH achieves smoother results and, when using a high-order kernel, a faster convergence rate, but at the cost of lower computational efficiency.

1. Introduction

With the continuous development of high-performance computer, computational fluid dynamics (CFD) has been recognized as a promising approach to solve a wide range of industrial problems, and also to augment the understanding of many longstanding flow problems, from microfluidics to hydrodynamics and hypersonics [1,2]. While classical mesh-based CFD methods have achieved great success, the generation of high-quality meshes remains a major challenge in particular for complex geometries used in practical applications. As an alternative, the meshless method has attracted considerable attentions owing to its numerical formulation is based on particles and independent of the topology defined by a mesh. As one typical example, smooth particle hydrodynamics (SPH), whose numerical approximations are based on Gaussian-like kernel function [3,4], has been widely applied in CFD [5], structural mechanics [6], and other scientific and engineering applications [7–9], when difficulties present for the classical mesh-base methods.

SPH can be formulated both in the Lagrangian and Eulerian frameworks for flow simulations. While the particle position is updated with velocity in Lagrangian SPH, it is fixed in Eulerian SPH. The former shows obvious advantages in simulating the flows associated with topology changes and involving material interfaces, e.g. violent free-surface flow [10], multi-phase flow [11] and fluid-structure interaction (FSI) with rigid or flexible structures [12,13]. While the Lagrangian particle introduces topological flexibility, it can lead to poor distribution, and hence, large numerical errors to be handled by elaborate particle regularization techniques [14–17]. With the compensation of topological flexibility, however, Eulerian SPH alleviates this problem as the particles are fixed and the initial optimal distribution is unchanged during the simulation [18,19]. Therefore, it is easier to obtain more uniform, or overall less numerical errors. For example, Noutcheuwa et al. [20], Lind et al. [21,22] and Wang et al. [23] have recently demonstrated high accuracy of Eulerian SPH for incompressible flows using a high-order smoothing kernel for interpolation. Another advantage of Eulerian SPH is that the com-

* Corresponding author.

E-mail addresses: zhentong.wang@tum.de (Z. Wang), c.zhang@tum.de (C. Zhang), oskar.haidn@tum.de (O.J. Haidn), nikolaus.adams@tum.de (N.A. Adams), xiangyu.hu@tum.de (X. Hu).

<https://doi.org/10.1016/j.cpc.2025.109683>

Received 9 November 2023; Received in revised form 1 April 2025; Accepted 15 May 2025

Available online 23 May 2025

0010-4655/© 2025 Published by Elsevier B.V.

putational efficiency can be much higher than that of its Lagrangian counterpart due to the fixed particles [24].

On the other hand, it is known in SPH community that the pairwise particle interaction using kernel-based particle formulation in SPH discretization can be considered as an analog of the numerical flux between the surface of two computational cells in the main-stream Eulerian mesh-based method, i.e. the finite volume method (FVM) [25,16,26]. Such analog has been detailed in Neuhauser [27] so that a FVM discretization is able to reuse the Riemann solver developed for the arbitrary-Eulerian-Lagrangian (ALE) SPH method in the same software package. Following this work, Wang et al. [28] proposed a new boundary algorithm that meets FVM's boundary requirements in the Eulerian SPH method for the improved computational efficiency and potential strong coupling. However, to which level can such analog, especially flux properties, reach between Eulerian SPH and FVM has not been explored yet. Furthermore, baring with the similarities and differences in the level of discretization and topology relations [27], the performance and computational efficiency between Eulerian SPH and mesh-based FVM is still unclear. To quantitatively assess this aspect, it is essential to implement both methods by achieving algorithmic equivalence and using a shared solver, including discretized conservation equations, zero-order consistency, flux properties, Riemann solutions, dissipation limiters and time integration, within the same codebase. Although many techniques have combined mesh-based and meshless methods [29–34] or be achieved in commercial softwares, they rely on their separate algorithms and solvers, with data transferred through a designed coupling method. To the best of our knowledge, no implementations enable the individual realization of both mesh-based and meshless methods using a shared solver within a unified framework.

In this paper, to achieve the rigorous comparison between these two methods, the first task is to introduce a form of the Eulerian SPH method where the kernel gradient matrix [35] is introduced to alleviate the constraints associated with the normal direction of interfaces, thus achieving the algorithmic equivalence to the FVM. Besides, the second task is that the traditional FVM formulation can be exactly implemented within an open-source library SPH, SPHinXsys [36], by developing a parser to extract necessary information from the “.msh” format mesh file generated from ICEM. By enabling a shared solver for both methods within a unified codebase, i.e. single algorithm for two methods, a rigorous comparison can be effectively performed. Note that the unstructured mesh in FVM is considered as it better accommodates complex geometries compared to the structured mesh [37,38]. In addition, to improve numerical stability and accuracy in both methods, the shared solver incorporates Riemann solutions [39,40], dissipation limiters [26,23], and dual-timestep integration [41].

This paper is structured as follows: in Section 2 the Riemann-based Eulerian SPH formulation together with the dissipation limiters is given. Additionally, we elaborate the kernel gradient correction matrix's role in the Eulerian SPH for achieving algorithmic equivalence with FVM. Besides, the detailed procedure for implementing mesh-based FVM within the framework of Eulerian SPH is presented. Rigorous comparisons on the performance between the Eulerian SPH with kernel gradient correction and FVM methods are given in Section 3 and Section 4 presents brief concluding remarks. All the computational codes employed in this study have been made openly accessible via the SPHinXsys repository [36,42], which can be accessed through the following URLs: <https://www.sphinxsys.org> and <https://github.com/Xiangyu-Hu/SPHinXsys>.

2. Methodology

In this section, the governing equations for fluid dynamics are briefly summarized and the corresponding discretizations for Riemann-based Eulerian SPH as well as the dissipation limiters are presented. Then, the FVM implementation within the framework of Eulerian SPH is detailed. Finally, the rigorous comparison between the Eulerian SPH with the kernel gradient correction and FVM is elaborated.

2.1. Governing equations

The Euler equation can be described by the following equation as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (1)$$

where $\mathbf{U}$ and $\mathbf{F}(\mathbf{U})$ are the vector of conserved variables and the corresponding fluxes, respectively. In two dimensional, they are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}, \quad (2)$$

respectively. Here, u and v are the components of velocity, ρ the density, p the pressure and $E = \frac{\rho v^2}{2} + \rho e$ the total energy with e being the internal energy. For compressible flow, we apply the equation of state (EOS)

$$p = \rho(\gamma - 1)e, \quad (3)$$

to close the system of Eq. (1). Here, γ is the heat capacity ratio and the speed of sound is given by

$$c^2 = \frac{\gamma p}{\rho}. \quad (4)$$

For incompressible flow, we follow the weakly compressible assumption by applying the artificial EOS

$$p = c^2(\rho - \rho_0). \quad (5)$$

Here, ρ_0 is the reference density. To limit the density variation within 1%, we set $c = 10U_{\max}$ with $U_{\max}$ denoting the maximum anticipated velocity of the flow field. Note that the energy conservation equation in Eq. (2) is neglected in the weakly compressible formulation. To simulate the viscous flow, a viscous force given by

$$f^{(v)} = \eta \nabla^2 \mathbf{v} \quad (6)$$

is added to the momentum equation.

2.2. Riemann-based Eulerian SPH method

In the present Eulerian SPH, the pairwise particle interaction is written in the flux form. Specifically, the mass, momentum and energy flux through the interface between a pair of particles is determined by solving an one-dimensional Riemann problem constructed along the interacting line $\mathbf{e}_{ij}$, as shown in Fig. 1 (left panel).

The left and right states of the Riemann problem are respectively defined as [26]

$$\begin{cases} (\rho_l, p_l, u_l) = (\rho_i, p_i, \mathbf{v}_i \cdot \mathbf{e}_{ij}) \\ (\rho_r, p_r, u_r) = (\rho_j, p_j, \mathbf{v}_j \cdot \mathbf{e}_{ij}) \end{cases}, \quad (7)$$

assuming that the discontinuity or interface is located at the middle point $\bar{\mathbf{r}}_{ij} = (\mathbf{r}_i + \mathbf{r}_j)/2$. Then, the Euler equation of Eq. (1) can be discretized as [25,26]

$$\begin{cases} \frac{\partial}{\partial t} (w_i \rho_i) + 2w_i \sum_j w_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (w_i \rho_i \mathbf{v}_i) + 2w_i \sum_j w_j \left[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbb{I} \right] \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (w_i E_i) + 2w_i \sum_j w_j \left[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^* \right] \cdot \nabla W_{ij} = 0 \end{cases} \quad (8)$$

Here, w is the volume of particle, $\mathbf{v}$ the velocity, $\mathbb{I}$ the identity matrix, $\nabla W_{ij} = \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}} \mathbf{e}_{ij}$ the gradient of kernel function with the unit vector $\mathbf{e}_{ij} = -\frac{\nabla W_{ij}}{|\nabla W_{ij}|}$. The terms $(\rho \mathbf{v})_{E,ij}^*$, $\left[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbb{I} \right]$ and $\left[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^* \right]$, representing mass, momentum and energy flux, respectively, are determined from the solution of Riemann problem. For

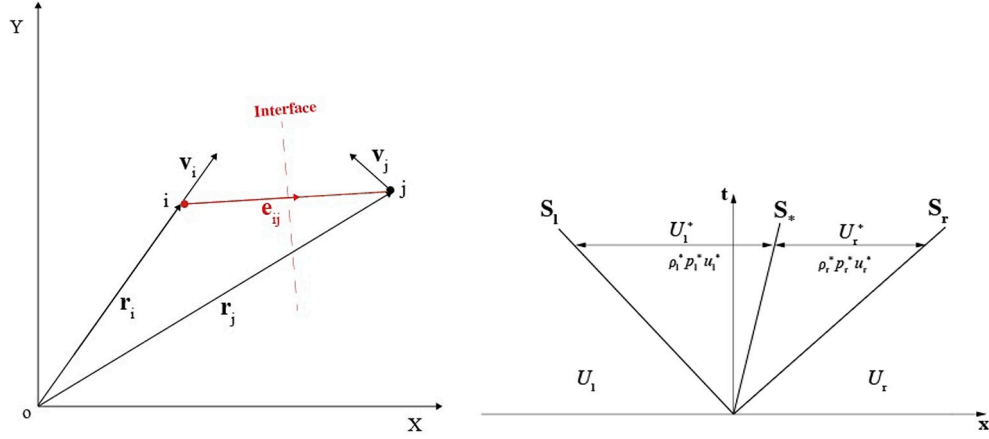


Fig. 1. Construction of Riemann problem along the interacting line of particles i and j where the interface is assumed passing the midpoint and the normal of that is along the interaction line (left panel); Simplified Riemann fan with two intermediate states (right panel).

the viscous flows, the viscous term in Eq. (6) can be discretized as follows:

$$f_i^{(v)} = 2 \sum_j m_j \frac{\eta}{\rho_i \rho_j} \frac{\mathbf{v}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}}, \quad (9)$$

with $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ and r_{ij} denoting the velocity difference and the distance between particles i and j , respectively.

The solution of the Riemann problem results in three waves emanating from the discontinuity, denoted by (ρ_l^*, u_l^*, p_l^*) and (ρ_r^*, u_r^*, p_r^*) as shown in Fig. 1 (right panel). Two waves, which can be shock or rarefaction wave, travel with the smallest wave speed S_l or largest wave speed S_r . The middle wave S_* is always a contact discontinuity and separates two intermediate states. Toro [39,40] has proposed the HLLC solver based on the HLL scheme [43] for more accurate and robust approximation of the Riemann problem for compressible fluid flows. In the HLLC scheme, the wave speeds S_l and S_r estimate for the left and right regions respectively, are

$$S_l = \bar{u} - \bar{c}, S_r = \bar{u} + \bar{c}, \quad (10)$$

where $\bar{u}$ and $\bar{c}$ denoting the average particle and sound speed detailed in Refs. [39,40]. Then, the intermediate wave speed S_* is calculated as

$$S_* = \frac{\rho_r u_r (S_r - u_r) + \rho_l u_l (u_l - S_l) + p_l - p_r}{\rho_r (S_r - u_r) + \rho_l (u_l - S_l)}. \quad (11)$$

Then other states in the star region can be derived as following

$$p^* = p_l + \rho_l (u_l - S_l) (u_l - u^*) = p_r + \rho_r (S_r - u_r) (u^* - u_r), \quad (12)$$

$$\mathbf{v}^* = u^* \mathbf{e}_{ij} + \left[\frac{1}{2} (\mathbf{v}_l + \mathbf{v}_r) - \frac{1}{2} (u_l + u_r) \mathbf{e}_{ij} \right], \quad (13)$$

$$\rho_{l/r}^* = \rho_{l/r} \frac{(S_{l/r} - u_{l/r})}{(S_{l/r} - u^*)}, \quad (14)$$

$$E_{l/r}^* = \frac{(S_{l/r} - u_{l/r}) E_{l/r} - p_{l/r} u_{l/r} + p^* u^*}{S_{l/r} - u^*}, \quad (15)$$

where $u^* = S_*$.

For weakly compressible fluid flows, with the assumption that the intermediate states satisfy $p_l^* = p_r^* = p^*$ and $u_l^* = u_r^* = u^*$, a linearized Riemann solver can be derived as [44]

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (16)$$

where $\bar{\rho}$ and $\bar{c}$ represent interface-particle averages. With the HLLC or linearized solution to the Riemann problem, the corresponding interface flux in Eq. (1) can subsequently be written as

$$F = \begin{cases} F_l & \text{if } S_l > 0 \\ F_l^* & \text{if } S_l \leq 0 < S_* \\ F_r^* & \text{if } S_* \leq 0 \leq S_r \\ F_r & \text{if } S_r < 0 \end{cases}. \quad (17)$$

Following Refs. [26,23], dissipation limiters are introduced for the HLLC and linearized Riemann solvers to decrease the numerical dissipation. In particular, we implement a low-dissipation HLLC Riemann solver, where the wave speed and pressure in the star region of Eqs. (11) and (12) are re-evaluated as

$$\begin{cases} u^* = \frac{\rho_l u_l c_l + \rho_r u_r c_r}{\rho_l c_l + \rho_r c_r} + \frac{p_l - p_r}{\rho_l c_l + \rho_r c_r} \beta_{HLLC}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{HLLC} [\rho_r c_r (u^* - u_r) - \rho_l c_l (u_l - u^*)] \end{cases}, \quad (18)$$

by introducing a dissipation limiter

$$\beta_{HLLC} = \min \left(\eta_{HLLC} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (19)$$

Note that we suggest that $\eta_{HLLC} = 1$ in Eulerian SPH framework and $\eta_{HLLC} = 10$ in FVM framework according to the numerical cases, and apply its squared value in the signal speed term for intensive dissipation control.

For the linearized Riemann solver, we also introduce a dissipation limiter to the both velocity and pressure terms and the linearized Riemann solver with the limiter [26] can then be derived as

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \beta_{linearisd}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{linearisd} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (20)$$

where the dissipation limiter $\beta_{linearisd}$ [26] is defined as

$$\beta_{linearisd} = \min \left(\eta_{linearisd} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (21)$$

Here, we suggest the parameter $\eta_{linearisd} = 15$ in both Eulerian SPH and FVM frameworks. Note that while many state-of-the-art techniques, such as high-order schemes [45–50], can achieve higher numerical accuracy, the differing implementations of these advanced techniques applied to Eulerian SPH and FVM make a rigorous comparison between these two methods challenging. Additionally, achieving higher accuracy using these advanced techniques is not our primary objective.

2.3. Comparison between Eulerian SPH and FVM

To understand the SPH formulation in Eulerian framework and its comparisons with FVM, we present a graphical illustration in two-dimensional between them in Fig. 2.

For the similarities between the both methods, they update the conserved variables by calculating the pairwise particle or cell interacting

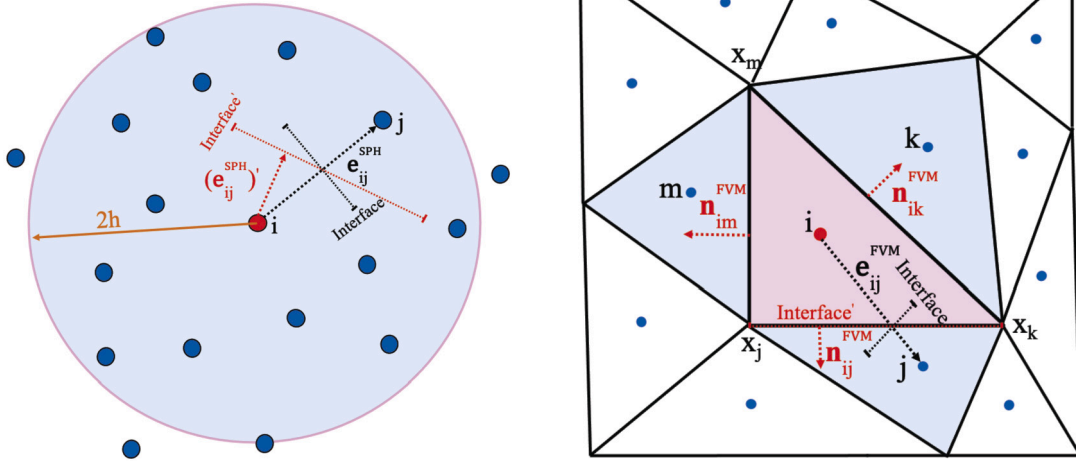


Fig. 2. Flux Integration: Over neighboring particles in SPH method (left panel); Over neighboring cells in finite volume method (right panel).

flux of all the neighbors. In addition, by analogy with the form of the SPH discretization Eq. (8), the both methods can be written uniformly as

$$\frac{\partial}{\partial t} (\omega_i \mathbf{U}_i) + \sum_j \mathbf{F}_{ij}(\mathbf{U}) \cdot \mathbf{A}_{ij} = 0. \quad (22)$$

Here, $\mathbf{A}_{ij}$ is a vector representing the interface area along the normal direction to the interface.

Also, we can compare the differences between the two methods in terms of the Eq. (22). The expressions of $\mathbf{A}_{ij}$ are different and denoted, respectively, as

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{A}_{ij}^{ESPH} = |\mathbf{A}_{ij}| \mathbf{n}_{ij}^{SPH} = 2\omega_i \omega_j \nabla W_{ij} = 2\omega_i \omega_j \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}} \mathbf{e}_{ij}^{SPH} & \text{in Eulerian SPH} \\ \mathbf{A}_{ij}^{FVM} = |\mathbf{A}_{ij}| \mathbf{n}_{ij}^{FVM} = S_{ij} \mathbf{n}_{ij}^{FVM} & \text{in FVM} \end{cases}, \quad (23)$$

where S_{ij} is the interface size between cell i and j , and the interface unit normal vector between particles i and j is $\mathbf{n}_{ij}^{SPH}$ in Eulerian SPH or cells i and j is $\mathbf{n}_{ij}^{FVM}$ in FVM. In SPHInXsys library, the gradient of the kernel ∇W_{ij} is stored separately as magnitude of gradient $\frac{\partial W_{ij}}{\partial r_{ij}}$ and displacement unit vector $\mathbf{e}_{ij}^{SPH}$ shown in Fig. 2 (left panel) and the displacement $\mathbf{r}_{ij}^{SPH} = |\mathbf{r}_{ij}^{SPH}| \mathbf{e}_{ij}^{SPH}$ is stored as the distance $|\mathbf{r}_{ij}^{SPH}|$ and displacement unit vector $\mathbf{e}_{ij}^{SPH}$. In FVM, we denote the center of the mesh as centroid and the displacement of centroids i and j $\mathbf{r}_{ij}^{FVM} = |\mathbf{r}_{ij}^{FVM}| \mathbf{e}_{ij}^{FVM}$ is along $\mathbf{e}_{ij}^{FVM}$, but the fluxes through the interface in Eq. (1) is along $\mathbf{n}_{ij}^{FVM}$ shown in Fig. 2 (right panel). Note that interface unit normal vector $\mathbf{n}_{ij}^{FVM}$ is independent with $\mathbf{e}_{ij}^{FVM}$ in FVM. Therefore, a clear difference between Eulerian SPH and FVM is that in former the interface normal vector $\mathbf{n}_{ij}^{SPH}$ must be aligned displacement direction $\mathbf{e}_{ij}^{SPH}$ [27]. A certain form of Eulerian SPH method is proposed later in section 2.4 to release this constraint between $\mathbf{n}_{ij}^{SPH}$ and $\mathbf{e}_{ij}^{SPH}$ and thus achieve the effect rigorously equivalent to FVM. Besides, as is shown in Fig. 2 and the Eq. (23), the interface area $|\mathbf{A}_{ij}|$ and the way to determine neighbors between cells in FVM are given from the known grid information (three or four neighboring cells in two or three dimensions, respectively), while those in SPH method is related to the gradient and the smoothing length h of the kernel (approximately 20 or 70 neighboring particles in two or three dimensions, respectively, when the kernel's cutoff radius is typically set as $2h$ where $h = 1.3dp$ with dp representing the initial spacing).

Furthermore, each cell as a completely closed control volume naturally obeys $\sum_j \mathbf{A}_{ij}^{FVM} = \mathbf{0}$ in FVM, while the original Eulerian SPH method with approximation errors leads to $\sum_j \mathbf{A}_{ij}^{ESPH} \approx \mathbf{0}$, i.e. with consistency error [27], which will be fixed by the particle relaxation and be further analyzed by coupling together with kernel gradient correction in section 2.4. Note that the similarities and differences between these two methods are general, while the specific flux properties depend on the particle distribution in the Eulerian SPH method and the mesh file information in FVM.

2.4. Kernel gradient correction

In the pursuit of a certain of Eulerian SPH method algorithmically equivalent to FVM where the direction $\mathbf{n}_{ij}$ is independent with $\mathbf{e}_{ij}$ discussed in Section 2.3, a crucial step is the introduction of the kernel gradient correction expressed as

$$\mathbf{B}_i = - \left(\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} w_j \right)^{-1}, \quad (24)$$

which has the properties of enhancing the numerical accuracy and meeting the first-order consistency as previously investigated [35,51]. Then the gradient of kernel can be rewritten as

$$\nabla' W_{ij} = \frac{\mathbf{B}_i + \mathbf{B}_j}{2} \nabla W_{ij} \quad (25)$$

to guarantee the momentum conservation. Note that while Eqs. (24) and (25) have also been implemented in Refs. [51,23,52,28] to decrease integration errors, the correction introduced here is specifically intended to release constraints, which has not been previously discussed [52,28].

As is mentioned above, interface unit normal vector $\mathbf{n}_{ij}^{SPH}$ in Eulerian SPH has to be along the displacement unit direction $\mathbf{e}_{ij}^{SPH}$. Based on this, the kernel correction matrix in Eq. (24) is implemented to release the constraint and to adjust the normal direction of interface along $(\mathbf{e}_{ij}^{SPH})'$ shown in Fig. 2 (left panel) expressed as

$$(\mathbf{e}_{ij}^{SPH})' = \frac{\nabla' W_{ij}}{|\nabla' W_{ij}|}. \quad (26)$$

Note that the interface between a pair of particles in Eulerian SPH is analogous to that between cells in FVM, but is virtual. Once the constraint is released, there exists a virtual mesh structure, which can be regarded as being generated by the SPH method and has the property of $(\mathbf{e}_{ij}^{SPH})'$ orthogonal to the corresponding virtual interface, analogous to the traditional mesh produced by mesh-generation softwares. Based

Table 1

Error analysis of SPH approximation for df/dx with a constant function $f = 1$ without and with the kernel gradient correction in the given relaxed particle distribution.

	Without-B	With-B
$L_1(df/dx)$	2.9693×10^{-4}	2.9636×10^{-4}

on this, $(\mathbf{e}_{ij}^{SPH})'$ is analogous to $\mathbf{n}_{ij}^{FVM}$ in traditional mesh-based FVM, thus making Eulerian SPH a certain version of FVM. Then the modified unit normal vector of interface $(\mathbf{e}_{ij}^{SPH})'$ is used to replace the original vector $\mathbf{e}_{ij}^{SPH}$ in Eq. (8).

It is worth noting that, in practical applications containing complex geometry, the lattice particle distribution is insufficient, so particle relaxation [53] given by

$$\mathbf{a}_{p,i} = -\frac{2}{\rho_i} \sum_j p_0 \nabla W_{ij} w_j \quad (27)$$

with p_0 denoting the constant pressure is implemented to make the particles fit precisely on the surface of the complex geometry, which also makes the particles satisfy the zero-order consistency [53], i.e. gradient of the constant function equal to zero. However, while the kernel gradient correction plays a pivotal role for Eulerian SPH method in achieving the FVM algorithm, it unavoidably impacts the zero-order consistency. Consequently, the incorporation of the kernel gradient correction becomes imperative for error analysis related to zero-order consistency. For the error analysis, L_1 normalization is given by

$$L_1(df/dx) = \frac{1}{N} \sum_i \left| \nabla f_i - 2 \sum_j \tilde{f}_{ij} \nabla W_{ij} w_j \right|, \quad (28)$$

where f is a function field and $2 \sum_j \tilde{f}_{ij} \nabla W_{ij} w_j$ represents the weak form [7] of SPH approximation with $\tilde{f}_{ij} = \frac{1}{2}(f_i + f_j)$ due to the conservation property in Eq. (8). In this case, we examine a circular geometry with a diameter of $D = 2$ using a resolution of $dp = D/20$. Employing a relaxed particle distribution based on the Wendland kernel [54], we conduct an error analysis of the SPH approximation for df/dx concerning a constant function $f = 1$ to assess zero-order consistency. The results in Table 1 present the L_1 error under two scenarios: one without and one with the utilization of kernel gradient correction, validating that employing the gradient kernel correction has a negligible effect on the approximation error compared to not using it. Consequently, Eulerian SPH, combined with kernel gradient correction, achieves algorithmic equivalence to the mesh-based FVM regarding the zero-order consistency, flux direction of the interface as well as flux weight from the neighbors.

2.5. Mesh-based FVM within Eulerian SPH framework

We have previously explored the utilization of a certain form of Eulerian SPH method to attain an algorithm equivalent to FVM and in this section we delve further into implementing the mesh-based FVM within the framework of the original Riemann-based Eulerian SPH method, all within the same SPH codebase. By constructing a parser, we read the external mesh file format generated by the commercial software ICEM to obtain all necessary information to implement mesh-based FVM in SPHinXsys. In the mesh file, the node positions and the topological relations of all meshes can be obtained directly, and then other required information including the size of the interface, its normal unit vector and the distance between centroids can further be calculated. Note that for cells adjoining the boundary surface in the mesh file, while topological relationships still include all neighboring cell indexes, one cell index is labeled as "0," signifying the boundary surface for all boundary conditions. This complicates the implementation of periodic boundary conditions in mesh-based FVM, as periodic information cannot be obtained from the mesh file.

In the original Eulerian SPH, we store the kernel gradient as interface unit normal vector $\mathbf{e}_{ij}^{SPH}$ and the magnitude of kernel gradient $|\nabla W_{ij}|$ separately mentioned in Section 2.3. Based on the relation between two methods in Eq. (23), it can be deduced that

$$\frac{\partial W_{ij}}{\partial r_{ij}} = \frac{S_{ij}}{2\omega_i \omega_j}. \quad (29)$$

To implement the traditional mesh-based FVM in SPH method, following the data structure of SPHinXsys, we analogize the storage form of FVM to SPH method as also two parts including the interface unit normal vector $\mathbf{n}_{ij}^{FVM}$ and the magnitude of kernel gradient $\frac{S_{ij}}{2\omega_i \omega_j}$ where $\mathbf{n}_{ij}^{FVM}$ and S_{ij} are calculated from the mesh information. Therefore, the flux properties of the both methods are stored in the same data structure. It is important to clarify that $\frac{S_{ij}}{2\omega_i \omega_j}$ in FVM, even stored and labeled as magnitude of the kernel gradient, actually is only determined by the interface size and is not related to the kernel, thus ensuring equivalence in the data structure and algorithms of SPHinXsys and facilitating the realization of both mesh-based FVM and SPH methods. Also, we store r_{ij}^{FVM} , the distance of centroids i and j , in SPHinXsys storage space r_{ij} which is used for solving the viscous force in Eq. (9). Following this procedure, the SPH code is endowed with the capability to simulate traditional mesh-based FVM, and the algorithmic equivalency between a certain form of Eulerian SPH and mesh-based FVM methods enables their seamless integration using the shared solver, including the flux data structure, Riemann solutions and dissipation limiters as well as the time integration, within the same SPH codebase, i.e. single algorithm for two methods. Furthermore, in the Eulerian SPH method, a wall boundary condition using the one-sided Riemann solver [55,26] is applied within the predefined solid particles. In addition, after calculating the distance between the centroid of the cell and the boundary surface from the mesh information, the one-sided Riemann solver is also introduced in the mesh-based FVM within the SPH code.

2.6. Time integration

For the time integration, we apply the Verlet scheme [13] where the total energy and density are first updated to the half time step $n + \frac{1}{2}$ by

$$\begin{cases} E_i^{n+\frac{1}{2}} = E_i^n + \frac{1}{2} \delta t \left(\frac{dE_i}{dt} \right)^n \\ \rho_i^{n+\frac{1}{2}} = \rho_i^n + \frac{1}{2} \delta t \left(\frac{d\rho_i}{dt} \right)^n \end{cases} \quad (30)$$

At this point, the internal energy and pressure are evaluated accordingly. Then, the change rate of momentum is calculated and applied to update the momentum to new time-step with

$$(\rho \mathbf{v})_i^{n+1} = (\rho \mathbf{v})_i^n + \delta t \left(\frac{d(\rho \mathbf{v})_i}{dt} \right)^{n+\frac{1}{2}}. \quad (31)$$

After that, the change rates of mass and energy are calculated. Finally, the energy and density for the new step are updated by

$$\begin{cases} E_i^{n+1} = E_i^{n+\frac{1}{2}} + \frac{1}{2} \delta t \left(\frac{dE_i}{dt} \right)^{n+\frac{1}{2}} \\ \rho_i^{n+1} = \rho_i^{n+\frac{1}{2}} + \frac{1}{2} \delta t \left(\frac{d\rho_i}{dt} \right)^{n+\frac{1}{2}} \end{cases} \quad (32)$$

In order to ensure numerical stability, the time step size is determined by

$$\Delta t_{ac} = CFL \left[\frac{L}{d(c_0 + U_{max})} \right], \quad (33)$$

where $CFL = 0.6$ in Eulerian SPH method and $CFL = 0.4$ in mesh-based FVM, d and U_{max} represent the dimension and the maximum particle speed in the fluid field, respectively. Here, L denotes the smoothing

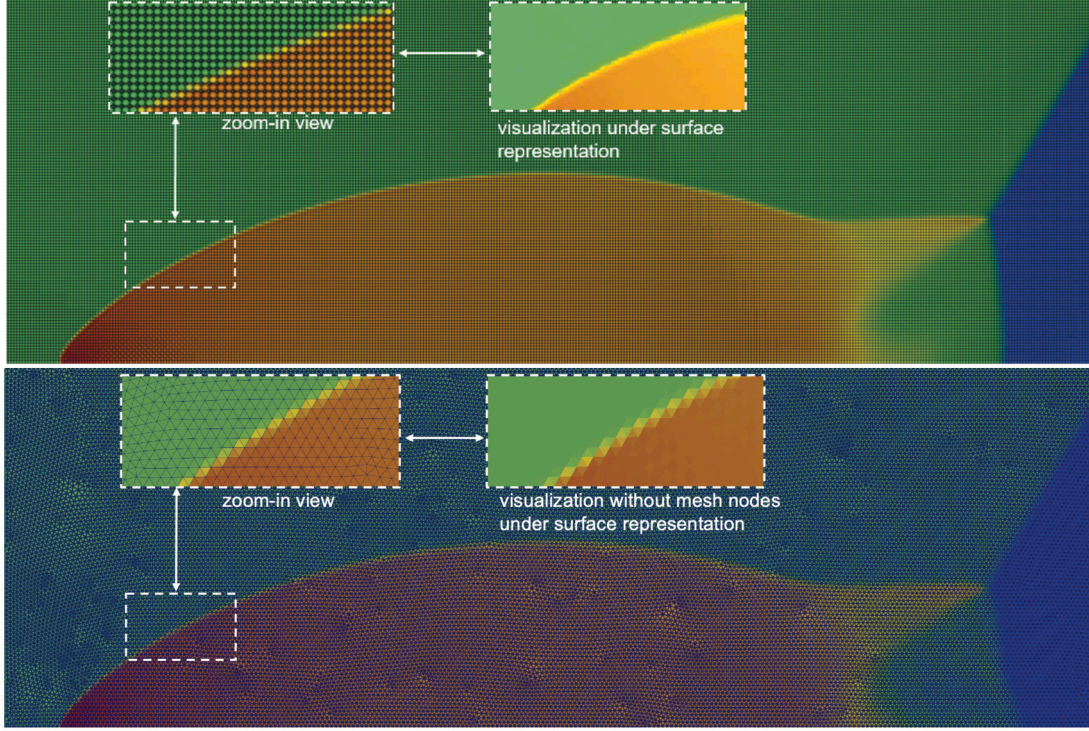


Fig. 3. Double Mach reflection of a strong shock: Density contour and its zoom-in view ranging from 1.3 to 23.0 obtained by ESPH (top panel) with the resolution $dp = 1/128$, i.e. the total particles number approximately $N = 6.5 \times 10^4$, and by FVM (bottom panel) with the total elements number approximately $N = 6.4 \times 10^4$ with the dissipation limiter at the final time $t = 0.2$.

length of kernel in Eulerian SPH or the minimum distance between mesh nodes in mesh-based FVM.

3. Numerical results

In this section, a set of numerical examples including both compressible and weakly compressible flows are considered herein to investigate the accuracy and stability of a certain form of Eulerian SPH method and its rigorous comparisons with mesh-based FVM. For all tests, the Laguerre-Gauss kernel [23] with a smoothing length of $h = 1.3dp$ is employed in the Eulerian SPH method due to its high numerical accuracy without requiring additional implementation and computational cost. For clarity, the Riemann-based Eulerian SPH method with the dissipation limiters and the kernel gradient correction is denoted as “ESPH”. Also, mesh-based FVM with the same algorithm including Riemann solvers and dissipation limiters is denoted as “FVM”. Note that we visualize the mesh information by changing the VTK file in SPHinXsys and the numerical results in Post-processing software called Paraview. In all the figures presented in the numerical examples, the zoom-in view figures with “under surface representation” indicate that they are displayed in a surface representation mode within Paraview. Here, we define the computational efficiency rate, denoted as α , which is the ratio of the CPU wall-clock times of “FVM” to “ESPH”, i.e., $\alpha = \frac{(\text{CPU wall-clock time})_{\text{FVM}}}{(\text{CPU wall-clock time})_{\text{ESPH}}}$. A lower α value indicates greater computational efficiency of using “FVM”.

3.1. Double Mach reflection of a strong shock

In this section, we test a two-dimensional problem namely double Mach reflection of a strong shock to rigorously compare ESPH with FVM in the compressible flow. Following Ref. [56], the computational domain is $(x, y) \in [0, 4] \times [0, 1]$ and the initial condition is given by

$$(\rho, u, v, p) = \begin{cases} (1.4, 0, 0, 1) & y \leq 1.732(x - 0.1667) \\ (8, 7.145, -4.125, 116.8333) & \text{otherwise} \end{cases},$$

the final time is $t = 0.2$. Besides, a right moving Mach 10 shock is initially located at $(x, y) = (0.1667, 0)$ and keeps a 60-degree angle with the x -axis. The bottom boundary is the reflective wall boundary beginning $x = 1/6$ to $x = 4$, the left-hand boundary is the post-shock boundary condition and zero-gradient condition is applied for the right boundary $x = 4$. In the case, the spatial resolution is $dp = 1/128$ with the total particles number approximately $N = 6.5 \times 10^4$ in ESPH and the maximum element seed size is 0.012 with the total elements number approximately $N = 6.4 \times 10^4$ in FVM to discretize the computational domain.

In Fig. 3, top panel is the density contour and its zoom-in view ranging from 1.3 to 23.0 obtained by ESPH with the resolution $dp = 1/128$ at the final time $t = 0.2$ and bottom panel presents that obtained by FVM with the total elements number approximately $N = 6.4 \times 10^4$ and the mesh visualization and its zoom-in view using FVM are also given. It can be observed that the main flow features including the Mach stem and the near-wall jet can be captured well in both methods. In the meantime, compared to the FVM, where unstructured meshes introduce anisotropic characteristics from three neighboring cells, ESPH exhibits a distinct advantage of producing smooth density contours due to its reliance on isotropic kernels and 20 neighboring particles, which naturally possess isotropic properties. Furthermore, to avoid the effect of the dissipation limiter on smoothness in FVM, we also calculate the density contour without using the limiter, as shown in Fig. 4. This demonstrates that even with excessive numerical dissipation, the density contour obtained by FVM is not as smooth as that obtained by the ESPH method.

3.2. Lid-driven cavity flows with different shapes

In this section, we consider two-dimensional lid-driven cavity flows with different shapes to compare ESPH with FVM further. Firstly, a simple square cavity is applied and the results calculated by both methods are compared with the results from Ghia et al. [57] to verify its correctness. Besides, a semi-circular cavity is tested to validate the capacity

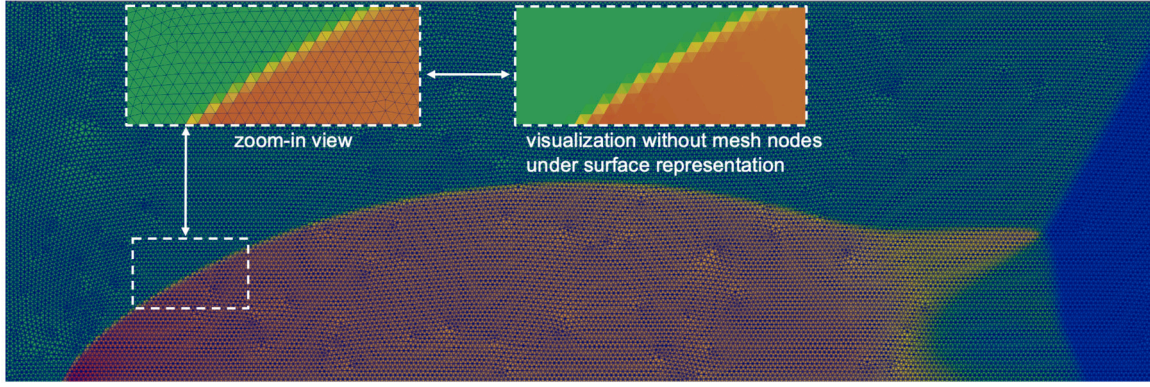


Fig. 4. Double Mach reflection of a strong shock: Density contour and its zoom-in view ranging from 1.3 to 23.0 obtained by FVM with the total elements number approximately $N = 6.4 \times 10^4$ without the dissipation limiter at the final time $t = 0.2$.

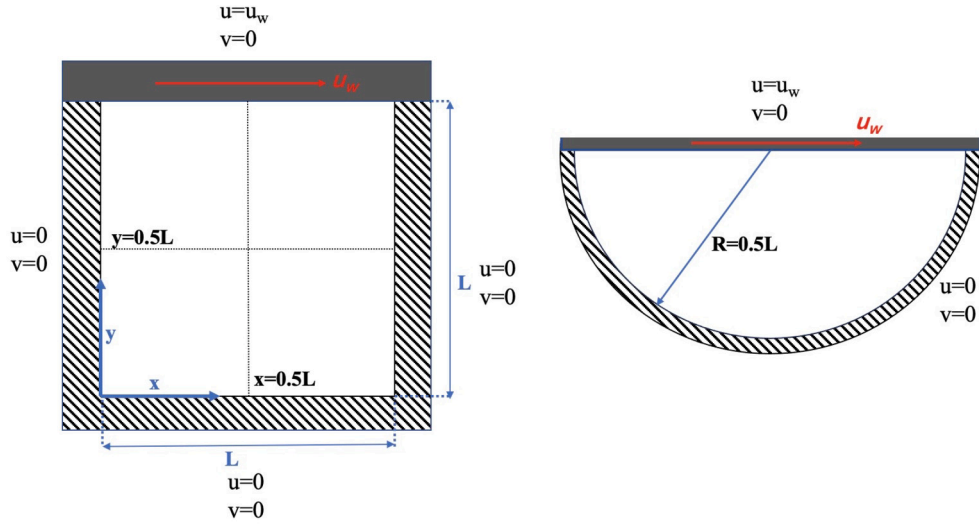


Fig. 5. Lid-driven cavity flows with different shapes: Geometries and boundary conditions.

to deal with complex geometry by comparing with the reference result from Glowinski et al. [58]. The geometries and boundary conditions are shown in Fig. 5 where the upper moving wall is set as a given velocity $U_{wall} = 1.0$ and other boundaries are non-slip wall conditions, and the final time $t = 30$.

3.2.1. Lid-driven square cavity problem

For the square cavity, the computational domain is a square with a length of $L = 1$ and the Reynolds number $Re = 400$ in the case. In ESPH, the spatial resolutions $dp = 1/33, 1/65$ and $1/129$ are applied with total particles numbers $N = 1089, 4225$ and 16641 , respectively, to verify the convergence study. Correspondingly, in FVM, the maximum element seed sizes are set as $0.05, 0.024$ and 0.012 with the total elements numbers $N = 1094, 4110$ and 16096 , respectively.

Fig. 6 presents the horizontal velocity component u along $x = 0.5L$ and the vertical velocity component v along $y = 0.5L$, obtained by FVM with total element numbers $N = 1094, 4110$, and 16096 , and ESPH with spatial resolutions of $dp = 1/33, 1/65$, and $1/129$, corresponding to total particle numbers $N = 1089, 4225$, and 16641 , respectively. Comparisons are made with reference results from Ghia et al. [57], indicating that the results are in good agreement with the reference data. Note that the convergence of the ESPH method using the high-order kernel is much faster than FVM, across resolutions ranging from low to high, clearly shown in right panel of Fig. 6. Additionally, Fig. 7 illustrates the horizontal and vertical velocity components obtained using both Eulerian SPH and FVM with and without (labeled as ‘ESPH-NL’ and ‘FVM-NL,’ respectively) limiters. The results indicate that the limiter substantially

reduces numerical dissipation in both methods, achieving greater reductions in FVM than in SPH. Besides, Fig. 7 also demonstrates the results converge even at higher resolutions.

For further quantitative error analysis, Fig. 8 illustrates the convergence rate of the L_1 -norm error at the extreme points, indicating that both methods attain approximately second-order convergence rate. As expected, the convergence rate at the extreme point is lower than the fourth order in ESPH [23] at low resolution as its much extra dissipation from the Riemann problem and limiters. Besides, Fig. 9 shows the velocity contour and its zoom-in view ranging from 2.0×10^{-5} to 0.97 obtained by ESPH with the total particles number $N = 16641$ and FVM with the total elements number $N = 16096$ under the Reynolds number $Re = 400$ in a square cavity, implying that ESPH enables obtaining smooth velocity contour while results obtained by FVM are not smooth shown in zoom-in figure of visualization without mesh nodes under surface representation. This is due to the same reason explained in the previous example, i.e. FVM with its unstructured mesh has anisotropic properties and Eulerian SPH relying on the isotropic kernel naturally has the isotropic characteristic. In the present study, the computations are all performed on an Intel Core i7-10700 2.90 GHz 8-core desktop computer and the total CPU wall-clock times required by ESPH with the total particles number $N = 16641$ and FVM with total elements number $N = 16096$ in whole process $t = 30$ are listed in Table 2, indicating that FVM is computationally much more efficient than ESPH with $\alpha = 0.387$. Note that the iteration number of FVM is significantly higher than that of Eulerian SPH due to the fact that the acoustic time step depends on the minimum distance between nodes in the mesh.

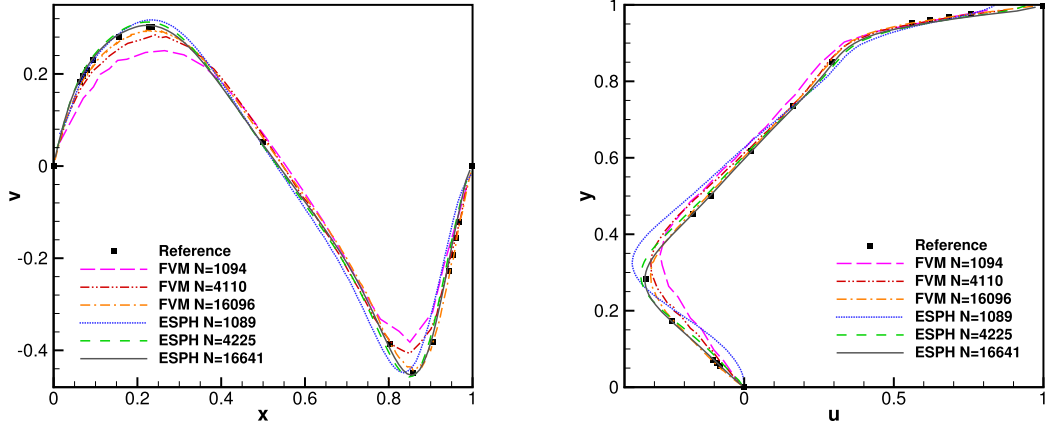


Fig. 6. Lid-driven square cavity flow with $Re = 400$: The horizontal velocity component u along $x = 0.5L$ (left panel) and the vertical velocity component v along $y = 0.5L$ (right panel) obtained by ESPH with the spatial resolutions as $dp = 1/33$, $1/65$ and $1/129$, i.e. the total particles numbers $N = 1089$, 4225 and 16641 , and by FVM with the total elements numbers $N = 1094$, 4110 and 16096 , as well as the comparisons with the reference obtained by Ghia et al. [57].

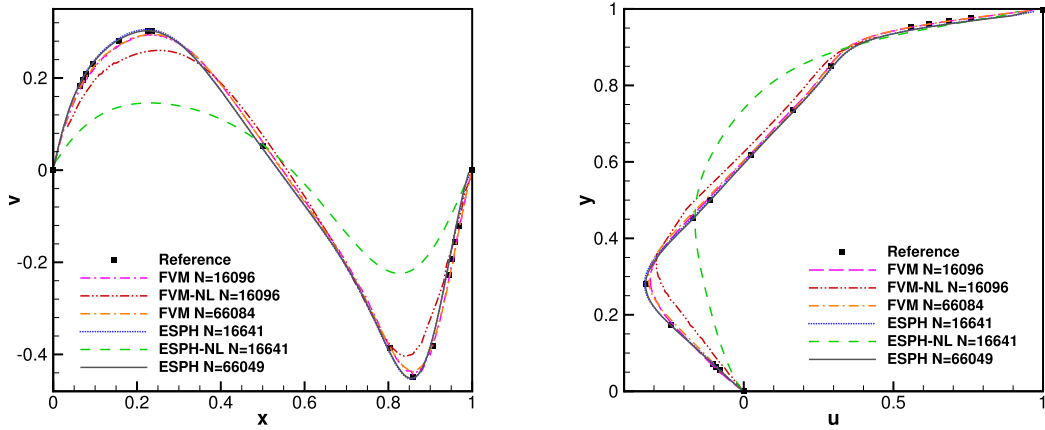


Fig. 7. Lid-driven square cavity flow with $Re = 400$: The horizontal velocity component u along $x = 0.5L$ (left panel) and the vertical velocity component v along $y = 0.5L$ (right panel) obtained by ESPH with $N = 16641$ and $N = 66049$, and FVM with total elements numbers $N = 16096$ and $N = 66084$ with and without the dissipation limiter, as well as the comparison with the reference obtained by Ghia et al. [57].

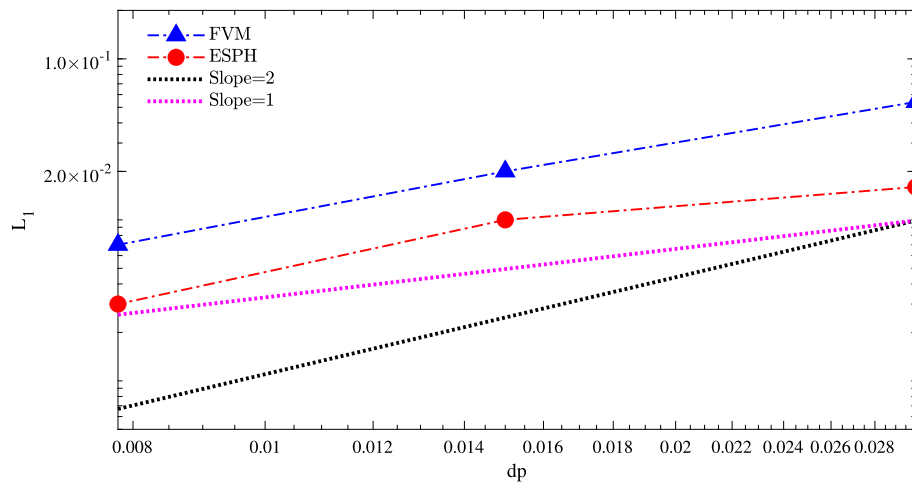


Fig. 8. Lid-driven square cavity flow with $Re = 400$: Convergence rate of L_1 -norm error of vertical velocity at the extreme point.

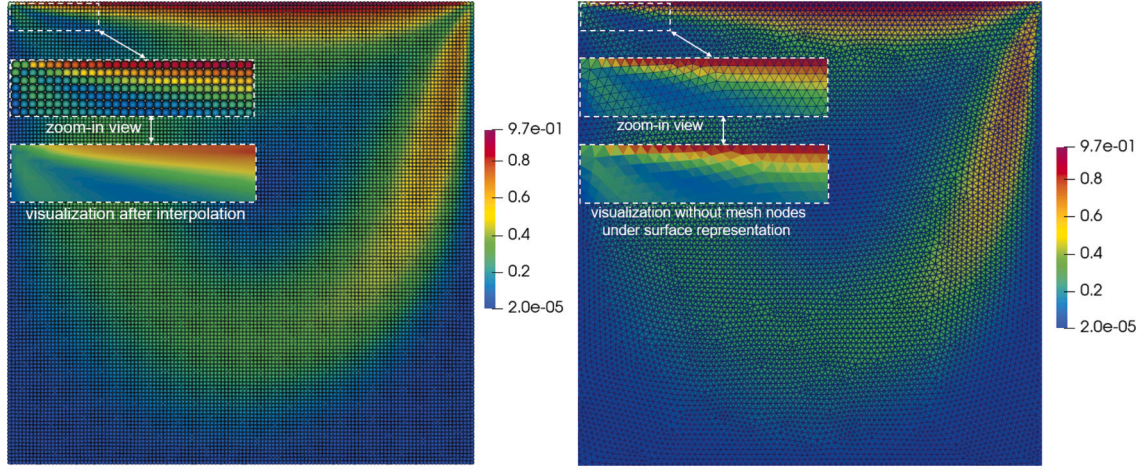


Fig. 9. Lid-driven square cavity flow with $Re = 400$: The velocity contour and its zoom-in view ranging from 2.0×10^{-5} to 0.97 obtained by ESPH with the total particles number $N = 16641$ (left panel) and FVM with the total elements number $N = 16096$ (right panel).

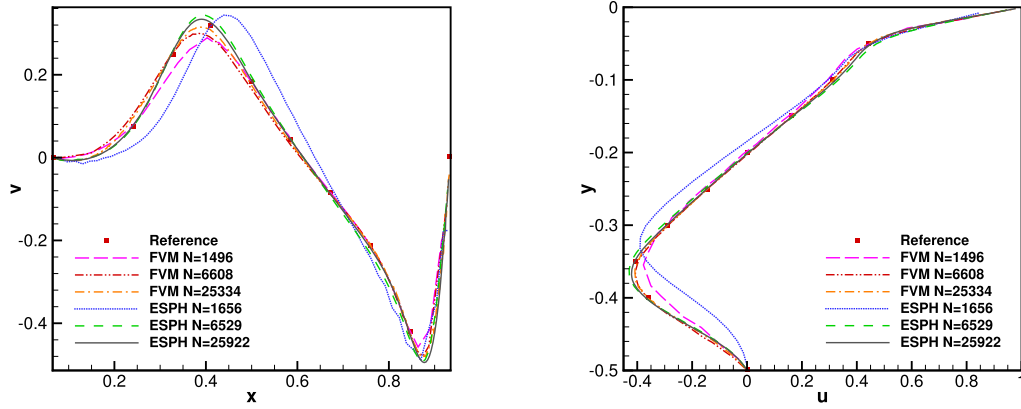


Fig. 10. Lid-driven semi-circular cavity flow with $Re = 1000$: The horizontal velocity component u along $x = 0.5L$ (left panel) and the vertical velocity component v along $y = -0.25L$ (right panel) obtained by ESPH with the resolutions as $dp = 1/65$, $1/129$ and $1/257$, i.e. the total particles number $N = 1656$, 6529 and 25922 respectively, and FVM with the total elements number $N = 1496$, 6608 and 25344 , and the comparison with the reference obtained by Glowinski et al. [58].

Table 2

Lid-driven square cavity flow with $Re = 400$: Computational efficiency comparison between FVM and ESPH for the physical time $t = 30$.

	Element/Particle number	CPU wall-clock time	Iteration number	α
FVM	16096	190.38 s	229625	0.387
ESPH	16641	491.53 s	108924	

Table 3

Lid-driven semi-circular cavity flow with $Re = 1000$: Computational efficiency comparison between FVM and ESPH for the physical time $t = 30$.

	Element/Particle number	CPU wall-clock time	Iteration number	α
FVM	6608	68.99 s	197677	0.394
ESPH	6529	174.78 s	108768	

3.2.2. Lid-driven semi-circular cavity problem

For semi-circular cavity, the diameter of cycle is $L = 1$ and the Reynolds number is $Re = 1000$ in the case, with the resolutions $dp = 1/65$, $dp = 1/129$ and $dp = 1/257$, i.e. the total particles number $N = 1656$, $N = 6529$ and $N = 25922$ respectively in ESPH or the total elements number $N = 1496$, $N = 6608$ and $N = 25344$ in FVM, respectively.

Fig. 10 shows the horizontal velocity component u along $x = 0.5L$ and the vertical velocity component v along $y = -0.25L$ obtained by ESPH and FVM as well as the comparison with the reference [58] in a semi-circular cavity, proving that both methods can obtain the accurate and consistent results which are agreement with the reference. Consistent with the findings from the previous example, the convergence rate of the ESPH method using the high-order kernel demonstrates a faster rate compared to the FVM, as shown in Fig. 10. Besides, Fig. 11 presents the velocity contour and its zoom-in view ranging from 2.0×10^{-5} to 0.97 obtained by ESPH with the spatial resolution $dp = 1/129$ and FVM with

the total elements number $N = 6608$, showing that, similarly with the square cavity results, the result calculated by ESPH is much smoother than that by FVM with its anisotropic characteristics in unstructured mesh-based method. Furthermore, for the computational efficiency analysis, the total CPU wall-clock times required by ESPH with the particle number $N = 6529$ and FVM with the total element number $N = 6608$ in whole process $t = 30$ are listed in Table 3, showing that the computational time cost in ESPH method is much longer.

3.3. Vortex spin-down

In the vortex spin-down simulation, the vortex is confined within a unit square bounded by four solid walls [59–61]. The flow field is initialized with a smooth, clockwise-rotating vortex defined as:

$$u = U_{max}(y - 0.5), v = U_{max}(0.5 - x), \quad (35)$$

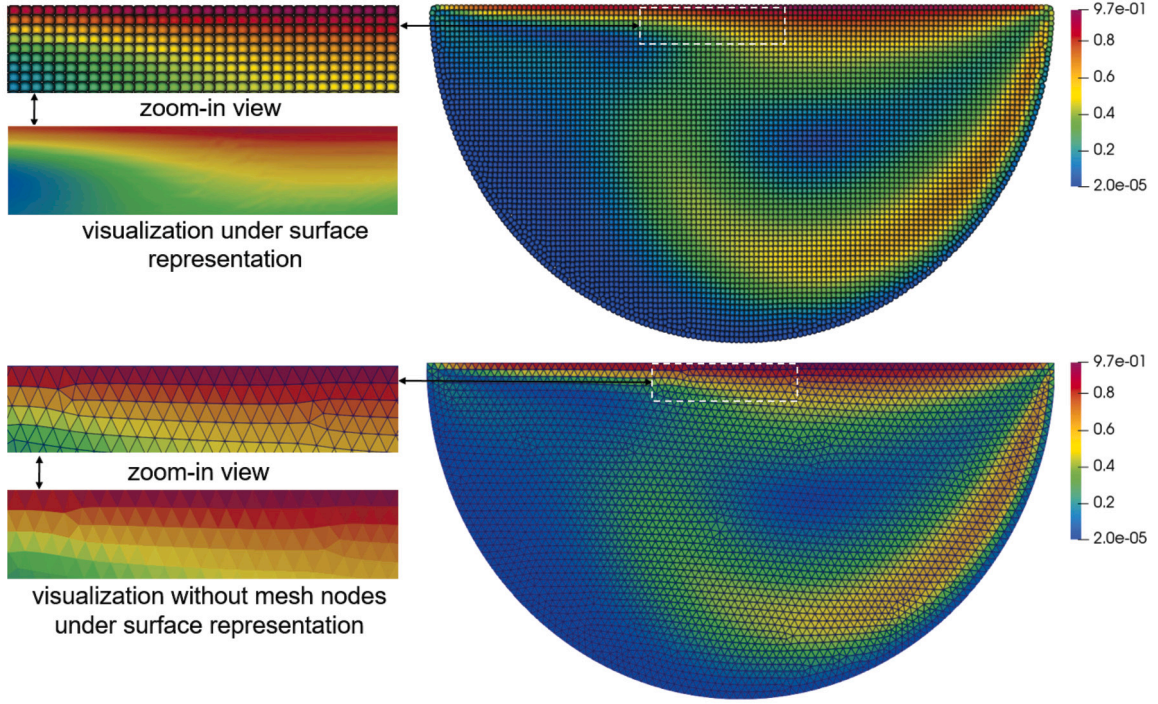


Fig. 11. Lid-driven semi-circular cavity flow with $Re = 1000$: The velocity contour and its zoom-in view ranging from 2.0×10^{-5} to 0.97 obtained by ESPH with the spatial resolution $dp = 1/129$, i.e. the total particles number $N = 6529$ (top panel), and FVM with the total elements number $N = 6608$ (bottom panel).

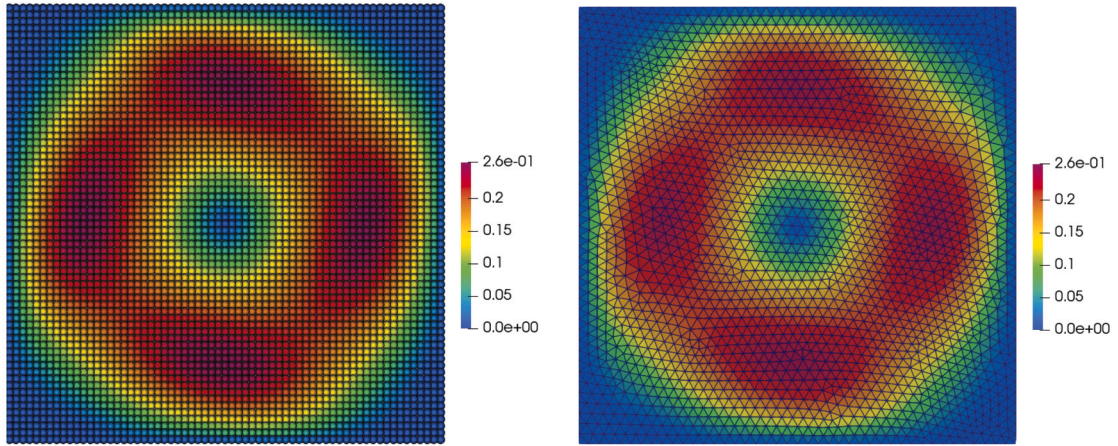


Fig. 12. Vortex spin-down with $Re = 100$: The velocity contours obtained by ESPH (left panel) and FVM (right panel) with $N = 4225$ and $N = 4110$, respectively, at $t = 1$.

where $U_{max} = 1$ represents the fluid maximum velocity. The Reynolds number for this case is set to $Re = 100$.

Fig. 12 displays the velocity contours ranging from 0 to 0.26 using ESPH with $N = 4225$ (left panel) and FVM with $N = 4110$ (right panel). Consistent with prior results, the Eulerian SPH method yields the smoother contour compared to FVM. For the quantitative validation, Fig. 13 (left panel) presents the vertical velocity profiles computed by ESPH and FVM with $N = 4225$ and $N = 4110$, respectively. Both methods achieve accuracy consistent with the benchmark result obtained by Xu et al. [60] for the comparable resolution. To further assess energy conservation and numerical dissipation, Fig. 13 (right panel) shows the kinetic energy decay over time for various resolutions using both methods, alongside a high-resolution solution labeled as “Reference”. The results confirm that while both methods exhibit convergence with particle/mesh refinement, the Eulerian SPH method demonstrates lower numerical dissipation as the results closer to the reference than FVM at the same resolution.

3.4. Flow around a circular cylinder

Furthermore, to rigorously compare ESPH and FVM in fluid-solid interaction, we investigate a case of flow around a circular cylinder as a benchmark case. For assessing numerical results quantitatively, the drag and lift coefficient are defined as

$$C_D = \frac{2F_D}{\rho_\infty u_\infty^2 A}, C_L = \frac{2F_L}{\rho_\infty u_\infty^2 A}, \quad (36)$$

where F_D and F_L are the drag and lift forces on the cylinder respectively. For unsteady cases, the Strouhal number $St = fD/u_\infty$ with f and D denoting the vortex shedding frequency and the cylinder diameter, respectively. In the case, the computational domain is $[40D, 40D]$ where the cylinder center is located at $(12.5D, 20D)$ and the Reynolds number $Re = u_\infty \rho_\infty D / \mu$ is 100 with the velocity $u_\infty = 1$, density $\rho_\infty = 1$ and the cylinder diameter $D = 2$. Besides, all boundary conditions are the far-field boundaries and the final time is 300.

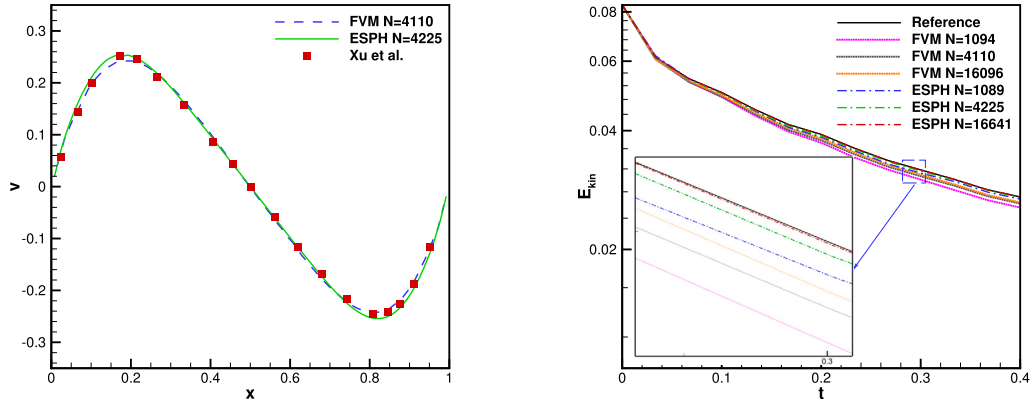


Fig. 13. Vortex spin-down at $Re = 100$: The vertical velocity component v along $y = 0.5$, computed using ESPH and FVM with $N = 4225$ and $N = 4110$, respectively, is compared against the similar solution from Xu et al. [60] at $t = 1$ (left panel). The right panel shows the kinetic energy decay over time for both methods at various resolutions, along with the reference.

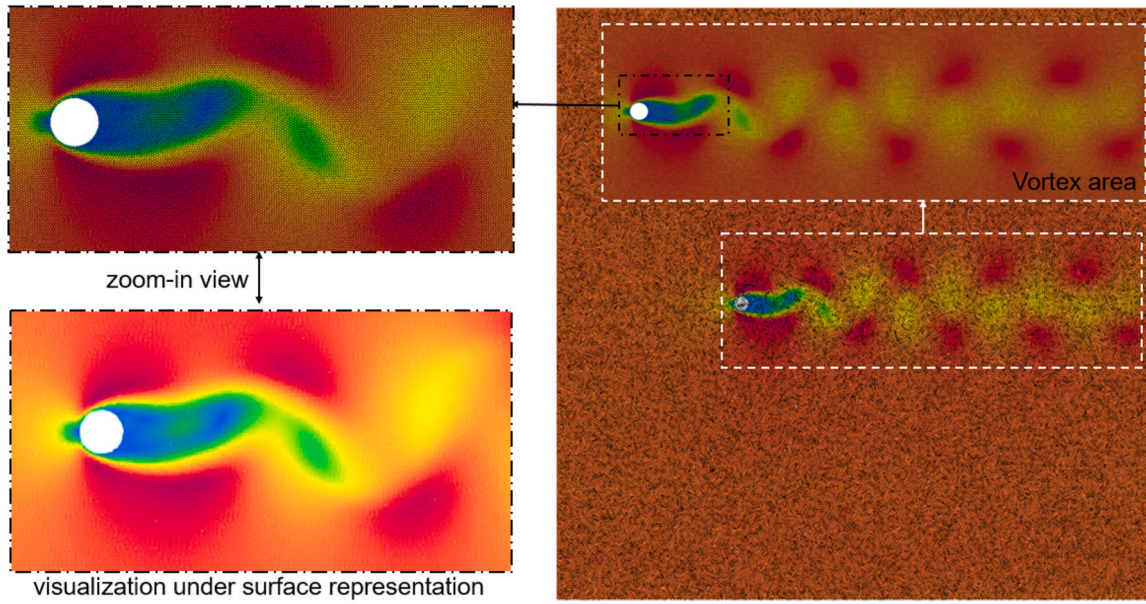


Fig. 14. Flow around a circular cylinder ($Re = 100$): Particle distribution and zoom-in view as well as velocity contour ranging from 0 to 1.32 obtained by ESPH with the spatial resolution $dp = 1/10$ with total particle number $N = 644492$ at the time $t = 300$.

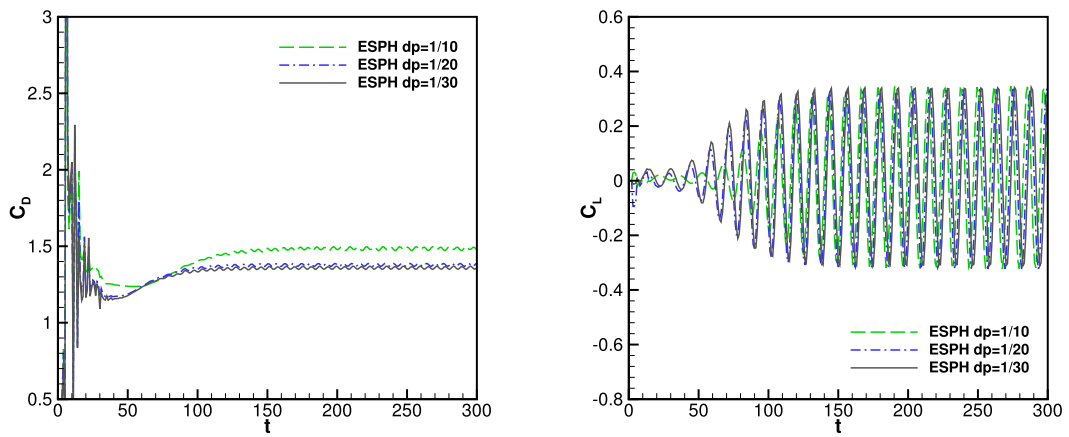


Fig. 15. Flow around a circular cylinder ($Re = 100$): Time evolution of the drag coefficient C_D (left panel) and lift coefficient C_L (right panel) obtained by ESPH with the spatial resolutions $dp = 1/10$, $1/20$ and $1/30$.

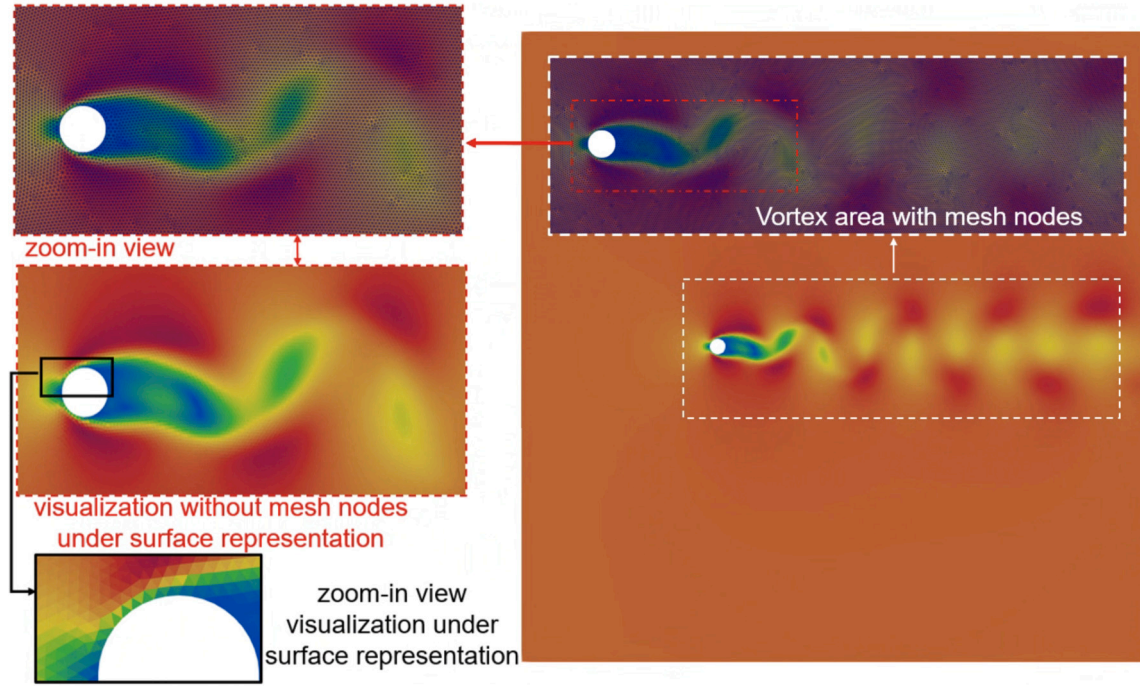


Fig. 16. Flow around a circular cylinder ($Re = 100$): Unstructured mesh and corresponding particle distribution as well as velocity contour obtained by FVM ranging from 0 to 1.32 with the total element number $N = 639543$ at the time $t = 300$.

Table 4

Flow around a circular cylinder ($Re = 100$): The converged drag and lift coefficients obtained by ESPH and FVM as well as other experimental and numerical results.

Parameters	C_D	C_L	S_i
White [62]	1.46	-	-
Chiu et al. [63]	1.35 ± 0.012	± 0.303	0.166
Le et al. [64]	1.37 ± 0.009	± 0.323	0.160
Brehm et al. [65]	1.32 ± 0.010	± 0.320	0.165
Russell et al. [66]	1.38 ± 0.007	± 0.300	0.172
Qu et al. [67]	1.32	± 0.225	0.165
ESPH	1.36 ± 0.011	± 0.340	0.167
FVM	1.36 ± 0.006	± 0.270	0.164

Table 5

Flow around a circular cylinder ($Re = 100$): Computational efficiency comparison between FVM and ESPH for the physical time $t = 300$.

	Element/Particle number	CPU wall-clock time	Iteration number	α
FVM	639543	6521.02 s	103267	0.554
ESPH	644492	11772.82 s	108768	

In ESPH, the spatial resolutions $dp = 1/10$, $1/20$ and $1/30$ are applied. Fig. 14 shows the particle distribution and the zoom-in view around the cylinder as well as velocity contour ranging from 0 to 1.32 obtained by ESPH with the resolution $dp = 1/10$ at the final time $t = 300$, indicating that the velocity distribution calculated by ESPH is very smooth because of its isotropic characteristics. Besides, Fig. 15 depicts the drag coefficient C_D and lift coefficient C_L obtained by ESPH with the spatial resolutions $dp = 1/10$, $1/20$ and $1/30$, showing that the drag coefficients reach a stable mean value after a period of fluctuation at the beginning while the lift coefficient oscillates around zero. The deviations of the drag and lift coefficients with the spatial resolutions $dp = 1/20$ and $dp = 1/30$ are less than 3 percent, and the frequencies and amplitudes of the lift coefficient are roughly consistent, which means that the results are convergent. The convergent results [63–68] are listed in Table 4 which contains other experimental and numerical, including SPH

and FVM, results under the Reynolds number $Re = 100$, showing that the results of the drag and lift coefficients obtained by ESPH agree well with other references and can be seen as correct.

Correspondingly, in FVM, the maximum element sizes are 0.15, 0.076 are applied with the total elements numbers are 639543 and 2534663, respectively. Fig. 16 portrays the visualization of unstructured mesh, velocity contour and its zoom-in view around the cylinder ranging from 0 to 1.32 obtained by FVM with the total elements number $N = 639543$ at the time $t = 300$. The velocity distribution appears generally smooth when observed globally, yet upon closer inspection in the zoomed-in visualization without mesh nodes, it reveals non-smoothness owing to the anisotropic property of the mesh-based FVM. Also, the convergent result obtained by FVM with the total elements number $N = 2534663$ is also listed in Table 4, implying that the drag and lift coefficients obtained by FVM are well agreement with the references. Furthermore, the total CPU wall-clock times required by ESPH with the particle number 644492 and FVM with the total element number $N = 639543$ in whole process are listed in Table 5, indicating the higher computational efficiency of the FVM.

3.5. Flow around the NACA5515 airfoil

In this section, more challenging scenario, flow around the NACA5515 airfoil with a sharp angle, is considered. The computational domain is defined as a square cavity, with the airfoil's chord length designated as $L = 1$. The left boundary is positioned two chord lengths upstream of the leading edge, while the upper and lower boundaries are three chord lengths from the airfoil surface [69]. The right boundary is located three chord lengths downstream of the trailing edge. The left and right boundaries are defined as inlet and outlet conditions, respectively, while the upper and lower boundaries are treated as far-field conditions. The airfoil surface is subjected to a no-slip boundary condition. The Reynolds number for the case is set as $Re = 420$ and simulation time is $t = 15$.

Fig. 17 displays the velocity contours, including zoom-in views with particle or mesh distributions, obtained using ESPH with total particle number $N = 2618626$ (left panel) and FVM with total element number $N = 2652573$ (right panel) at $t = 15$, demonstrating that the former

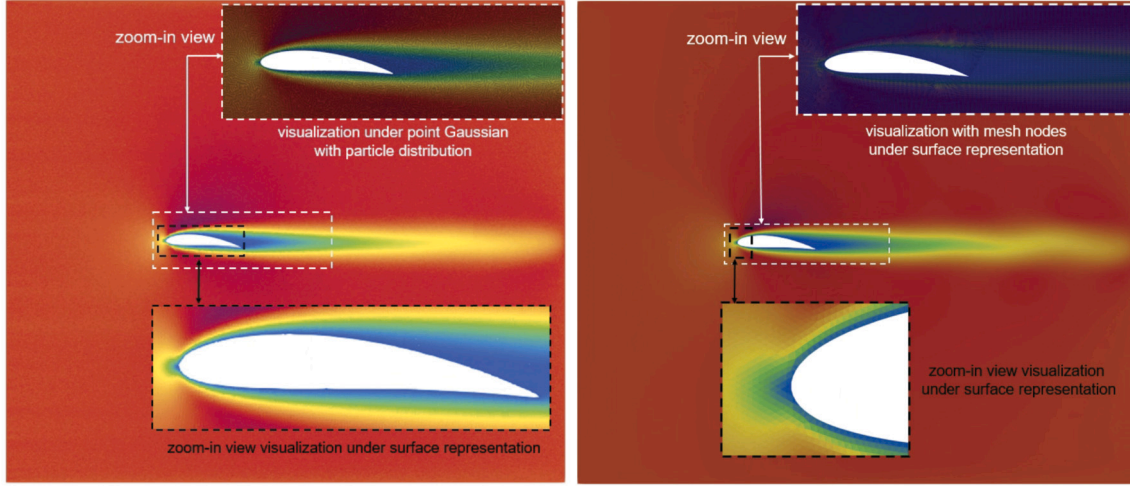


Fig. 17. Flow around the NACA5515 airfoil ($Re = 420$): Velocity contours and their zoom-in views ranging from 0 to 1.2 obtained by ESPH (left panel) with the resolution $dp = 1/250$, i.e. total particle number $N = 2618626$, and by FVM (right panel) with total element number $N = 2652573$ at the time $t = 15$.

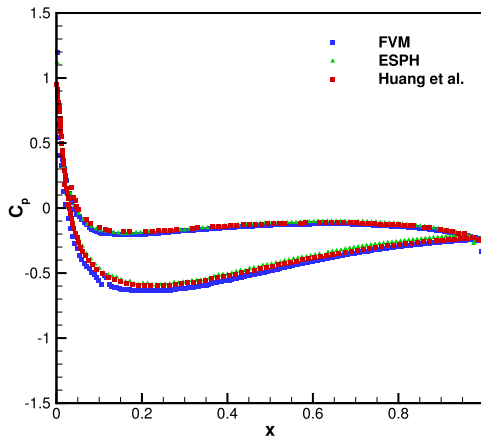


Fig. 18. Flow around the NACA5515 airfoil ($Re = 420$): Pressure coefficients on the surface obtained by ESPH with total particle number $N = 2618626$ and FVM with total element number $N = 2652573$ at the time $t = 15$.

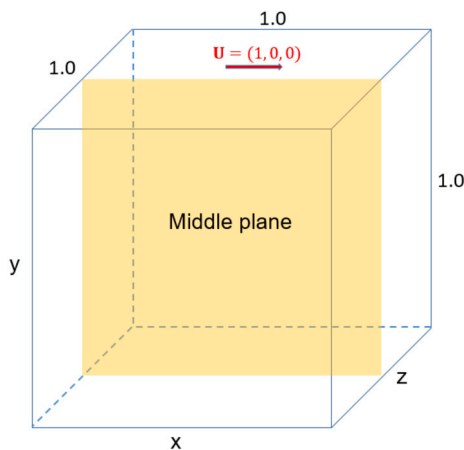


Fig. 19. 3D lid-driven cavity flow: Geometries and boundary conditions.

yields the smoother flow contour compared to the latter. Furthermore, Fig. 18 presents the pressure coefficients on the surface using ESPH and FVM as well as the numerical result by Huang et al. [69], showing both methods achieve the accurate results even in the challenging scenario.

Table 6

3D lid-driven cavity flow with $Re = 100$: Computational efficiency comparison between FVM and ESPH for the physical time $t = 10$.

	Element/Particle number	CPU wall-clock time	Iteration number	α
FVM	128418	589.60 s	49803	0.19
ESPH	132651	3113.95 s	21862	

3.6. 3D lid-driven cavity flow

This section presents a further comparison of the performance of ESPH and FVM in a three-dimensional case, namely 3D lid-driven square cavity flow, using a cube of unit length. Following Refs. [70,71], similar to the 2D lid-driven cavity flow, the upper surface is designed as a moving boundary with $U = (1.0, 0.0, 0.0)$, while the remaining surfaces are set as stationary non-slip boundaries, as shown in Fig. 19. In this case, the Reynolds number is $Re = 100$ and the final time is $t = 10$, with the total particles $N = 132651$, 1030301 and 2146289 in ESPH or total elements $N = 128418$, 1031317 and 2108628 in FVM to discretize the computational domain.

Fig. 20 presents the velocity contour of the block after the middle plane using ESPH and FVM, alongside the reference result from [70]. This validates that the velocity contour obtained by ESPH is smoother than that obtained by FVM. Additionally, Fig. 21 shows the velocity profiles of the horizontal and vertical centerlines of the middle plane using ESPH with the total particle number of $N = 2146289$ and FVM with the total element count of $N = 2108628$, proving the results using both methods are consistent with the reference results and proving that the mesh-based FVM realization in SPH code in three dimensional situation is also convincing and accurate. Besides, consistent with the findings from the two-dimensional cases, the convergence rate of the ESPH method using the high-order kernel is significantly faster compared to that of the FVM, as shown in Fig. 21. Furthermore, we test the computational efficiency of both methods. The total CPU wall-clock time required by ESPH with $N = 132651$ particles and FVM with $N = 128418$ elements in whole process is listed in Table 6, demonstrating that the higher computational efficiency of the FVM in 3D even significantly higher 2D.

4. Summary and conclusion

In this paper, the kernel gradient correction matrix is implemented to release the constraint on the Eulerian SPH method for adjusting flux

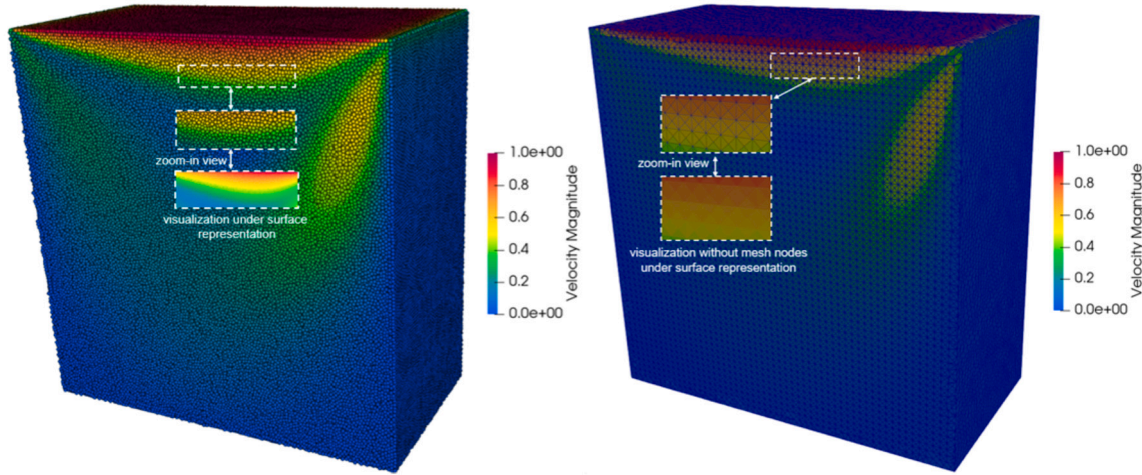


Fig. 20. 3D lid-driven cavity flow with $Re = 100$: The velocity contour and its zoom-in view of the block after the middle plane ranging from 0.0 to 1.0 obtained by ESPH with the total particles number $N = 2146289$ (left panel) and FVM with the total elements number $N = 2108628$ (right panel).

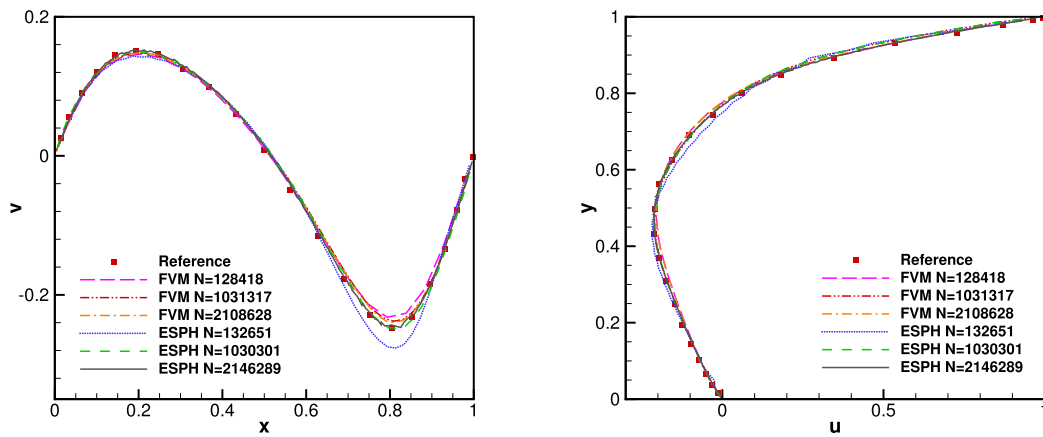


Fig. 21. 3D lid-driven cavity flow with $Re = 100$: The horizontal (a) and vertical (b) centerlines of the middle plane. Here, the results are obtained by ESPH with the total particles number $N = 132651$, 1030301 and 2146289, and FVM with the total elements number $N = 128418$, 1031317 and 2108628 and the comparison with the reference obtained by Ku et al. [70].

properties. After comparing Eulerian SPH and FVM, the algorithmic equivalence including the zero-order consistency, flux direction of the interface and flux weight from the neighbors can be achieved. Besides, the mesh-based FVM is realized in the same SPH program SPHnXsys using the shared solver, including the consistent discretized conservation equation, flux data structure, Riemann solutions, dissipation limiters and time integration scheme, and therefore both methods' computational efficiency and performance can be compared rigorously. Several fully and weakly compressible examples are tested, and the results have validated that while the mesh-based FVM has higher computational efficiency due to significantly fewer neighbors, the Eulerian SPH has a clear advantage in producing smoother outcomes owing to the kernel's isotropic properties and in achieving a faster convergence rate with the employment of the high-order kernel. It is crucial to acknowledge that the computational time and iteration number using the mesh-based FVM is significantly affected by the mesh property, as the time step is determined by the minimum distance between grid nodes. For future work, the rigorous comparison between these two methods will be extended to high-order schemes such as the implementation of slope limiters [72,73] and periodic boundary by developing efficient identification algorithms in FVM. Additionally, the strong coupling of the Eulerian SPH and FVM methods with a shared solver within a unified framework will be explored, providing enhanced stability and computational efficiency.

CRediT authorship contribution statement

Zhentong Wang: Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Chi Zhang:** Writing – review & editing, Investigation. **Oskar J. Haidn:** Writing – review & editing, Supervision, Investigation. **Nikolaus A. Adams:** Writing – review & editing, Investigation. **Xiangyu Hu:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] F. Afshari, H.G. Zavaragh, B. Sahin, R.C. Grifoni, F. Corvaro, B. Marchetti, F. Polonara, On numerical methods; optimization of cfd solution to evaluate fluid flow around a sample object at low re numbers, *Math. Comput. Simul.* 152 (2018) 51–68.

- [2] J. O'connor, J.M. Domínguez, B.D. Rogers, S.J. Lind, P.K. Stansby, Eulerian incompressible smoothed particle hydrodynamics on multiple gpus, *Comput. Phys. Commun.* 273 (2022) 108263.
- [3] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (3) (1977) 375–389.
- [4] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (1977) 1013–1024.
- [5] J.J. Monaghan, Simulating free surface flows with sph, *J. Comput. Phys.* 110 (2) (1994) 399–406.
- [6] L.D. Libersky, A.G. Petschek, Smooth particle hydrodynamics with strength of materials, in: *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*, Springer, 1991, pp. 248–257.
- [7] C. Zhang, Y. Zhu, D. Wu, X. Hu, Review on smoothed particle hydrodynamics: methodology development and recent achievement, *arXiv preprint, arXiv:2205.03074*, 2022.
- [8] M. Luo, A. Khayyer, P. Lin, Particle methods in ocean and coastal engineering, *Appl. Ocean Res.* 114 (2021) 102734.
- [9] H. Gotoh, A. Khayyer, Y. Shimizu, Entirely Lagrangian meshfree computational methods for hydroelastic fluid-structure interactions in ocean engineering—reliability, adaptivity and generality, *Appl. Ocean Res.* 115 (2021) 102822.
- [10] M. Gomez-Gesteira, B.D. Rogers, R.A. Dalrymple, A.J. Crespo, State-of-the-art of classical sph for free-surface flows, *J. Hydraul. Res.* 48 (sup1) (2010) 6–27.
- [11] M. Rezavand, C. Zhang, X. Hu, A weakly compressible sph method for violent multiphase flows with high density ratio, *J. Comput. Phys.* 402 (2020) 109092.
- [12] C. Antoci, M. Gallati, S. Sibilla, Numerical simulation of fluid–structure interaction by sph, *Comput. Struct.* 85 (11–14) (2007) 879–890.
- [13] C. Zhang, M. Rezavand, X. Hu, A multi-resolution sph method for fluid-structure interactions, *J. Comput. Phys.* 429 (2021) 110028.
- [14] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *Int. J. Numer. Methods Eng.* 66 (13) (2006) 2064–2085.
- [15] S. Adami, X. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, *J. Comput. Phys.* 241 (2013) 292–307.
- [16] S. Litvinov, X. Hu, N.A. Adams, Towards consistence and convergence of conservative sph approximations, *J. Comput. Phys.* 301 (2015) 394–401.
- [17] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B.D. Rogers, A. Souto-Iglesias, Grand challenges for smoothed particle hydrodynamics numerical schemes, *Comput. Part. Mech.* 8 (3) (2021) 575–588.
- [18] M. Basa, N.J. Quinlan, M. Lastiwka, Robustness and accuracy of sph formulations for viscous flow, *Int. J. Numer. Methods Fluids* 60 (10) (2009) 1127–1148.
- [19] A. Nasar, B.D. Rogers, A. Revell, P. Stansby, S. Lind, Eulerian weakly compressible smoothed particle hydrodynamics (sph) with the immersed boundary method for thin slender bodies, *J. Fluids Struct.* 84 (2019) 263–282.
- [20] R.K. Nouthuewa, R.G. Owens, A new incompressible smoothed particle hydrodynamics-immersed boundary method, *Int. J. Numer. Anal. Model. Ser. B* 3 (2) (2012) 126–167.
- [21] S. Lind, P. Stansby, Investigations into higher-order incompressible sph, in: *Proc. 10th SPHERIC Int. Workshop*, 2015, pp. 131–138.
- [22] S.J. Lind, P. Stansby, High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion, *J. Comput. Phys.* 326 (2016) 290–311.
- [23] Z. Wang, B. Zhang, O.J. Haidn, X. Hu, A fourth-order kernel for improving numerical accuracy and stability in Eulerian sph for fluids and total Lagrangian sph for solids, *J. Comput. Phys.* (2024) 113385.
- [24] A.M. Nasar, Eulerian and Lagrangian Smoothed Particle Hydrodynamics as Models for the Interaction of Fluids and Flexible Structures in Biomedical Flows, The University of Manchester, United Kingdom, 2016.
- [25] J. Vila, On particle weighted methods and smooth particle hydrodynamics, *Math. Models Methods Appl. Sci.* 9 (02) (1999) 161–209.
- [26] C. Zhang, X. Hu, N.A. Adams, A weakly compressible sph method based on a low-dissipation Riemann solver, *J. Comput. Phys.* 335 (2017) 605–620.
- [27] M. Neuhauser, Development of a coupled sph-ale/finite volume method for the simulation of transient flows in hydraulic machines, Ph.D. thesis, Ecole Centrale de Lyon, Ecully, 2014.
- [28] Z. Wang, O.J. Haidn, X. Hu, An algorithm for the incorporation of relevant fvm boundary conditions in the Eulerian sph framework, *Comput. Phys. Commun.* 307 (2025) 109429.
- [29] S. Liu, G. Yang, Y. Xu, D. Hu, A sph-fvm coupling method based on triangular mesh for the simulation of two-phase flows, *Eng. Anal. Bound. Elem.* 159 (2024) 237–258.
- [30] Y. Zhao, L. Huo, W. Xu, H. Zhu, Coupling of smoothed particle hydrodynamics and finite volume method for electrohydrodynamic droplet deformation simulation, *Comput. Fluids* 270 (2024) 106162.
- [31] Z. Wang, D. Zheng, X. Guo, Z. Gu, Y. Shen, T. Nian, Investigation of offshore landslides impact on bucket foundations using a coupled sph–fem method, *Geoenviron. Disasters* 11 (1) (2024) 2.
- [32] D. Hietel, K. Steiner, J. Struckmeier, A finite-volume particle method for compressible flows, *Math. Models Methods Appl. Sci.* 10 (09) (2000) 1363–1382.
- [33] N.J. Quinlan, L. Lobovsky, R.M. Nestor, Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area, *Comput. Phys. Commun.* 185 (6) (2014) 1554–1563.
- [34] D.A. Barcarolo, D. Touzé, G. Oger, F. de Vuyst, Voronoi-sph: on the analysis of a hybrid finite volumes-smoothed particle hydrodynamics method, in: *9th International SPHERIC Workshop Paris, France*, 2014.
- [35] P. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, *Comput. Methods Appl. Mech. Eng.* 139 (1–4) (1996) 375–408.
- [36] C. Zhang, M. Rezavand, Y. Zhu, Y. Yu, D. Wu, W. Zhang, J. Wang, X. Hu, Sphinxsys: an open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics, *Comput. Phys. Commun.* (2021) 108066.
- [37] M.A. Park, A. Loseille, J. Krakos, T.R. Michal, J.J. Alonso, Unstructured grid adaptation: status, potential impacts, and recommended investments towards cfd 2030, in: *46th AIAA Fluid Dynamics Conference*, 2016, p. 3323.
- [38] Y. Weng, X. Zhang, X. Guo, X. Zhang, Y. Lu, Y. Liu, Effects of mesh loop modes on performance of unstructured finite volume gpu simulations, *Adv. Aerodyn.* 3 (2021) 1–23.
- [39] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the hll-riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [40] E.F. Toro, The hllc Riemann solver, *Shock Waves* 29 (8) (2019) 1065–1082.
- [41] C. Zhang, M. Rezavand, X. Hu, Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics, *J. Comput. Phys.* 404 (2020) 109135.
- [42] C. Zhang, M. Rezavand, Y. Zhu, Y. Yu, D. Wu, W. Zhang, S. Zhang, J. Wang, X. Hu, Sphinxsys: an open-source meshless, multi-resolution and multi-physics library, *Softw. Impacts* 6 (2020) 100033.
- [43] A. Harten, P.D. Lax, B.v. Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Rev.* 25 (1) (1983) 35–61.
- [44] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer Science & Business Media, 2013.
- [45] A.M. Nasar, G. Fourtakas, S.J. Lind, B.D. Rogers, P. Stansby, J. King, High-order velocity and pressure wall boundary conditions in Eulerian incompressible sph, *J. Comput. Phys.* 434 (2021) 109793.
- [46] A. Vergnaud, G. Oger, D. Le Touzé, Investigations on a high order SPH scheme using WENO reconstruction, *J. Comput. Phys.* 477 (2023) 111889.
- [47] L. Fu, Review of the high-order teno schemes for compressible gas dynamics and turbulence, *Arch. Comput. Methods Eng.* 30 (4) (2023) 2493–2526.
- [48] Z.-F. Meng, P.-N. Sun, Y. Xu, P.-P. Wang, A.-M. Zhang, High-order Eulerian sph scheme through w/teno reconstruction based on primitive variables for simulating incompressible flows, *Comput. Methods Appl. Mech. Eng.* 427 (2024) 117065.
- [49] W.-K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (I) methodology and convergence, *Comput. Methods Appl. Mech. Eng.* 143 (1–2) (1997) 113–154.
- [50] G.A. Dils, Moving-least-squares-particle hydrodynamics—I. Consistency and stability, *Int. J. Numer. Methods Eng.* 44 (8) (1999) 1115–1155.
- [51] V. Zago, L.J. Schulze, G. Bilotta, N. Almasan, R. Dalrymple, Overcoming excessive numerical dissipation in sph modeling of water waves, *Coast. Eng.* 170 (2021) 104018.
- [52] Z. Wang, C. Zhang, O.J. Haidn, X. Hu, An efficient truncation scheme for Eulerian and total Lagrangian smoothed particle hydrodynamics methods, *Phys. Fluids* 36 (7) (2024).
- [53] Y. Zhu, C. Zhang, Y. Yu, X. Hu, A cad-compatible body-fitted particle generator for arbitrarily complex geometry and its application to wave-structure interaction, *J. Hydrodyn.* 33 (2) (2021) 195–206.
- [54] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.* 4 (1) (1995) 389–396.
- [55] S. Adami, X.Y. Hu, N.A. Adams, A generalized wall boundary condition for smoothed particle hydrodynamics, *J. Comput. Phys.* 231 (21) (2012) 7057–7075.
- [56] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.
- [57] U. Ghia, K.N. Ghia, C. Shin, High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.* 48 (3) (1982) 387–411.
- [58] R. Glowinski, G. Guidoboni, T.-W. Pan, Wall-driven incompressible viscous flow in a two-dimensional semi-circular cavity, *J. Comput. Phys.* 216 (1) (2006) 76–91.
- [59] S.J. Cummins, M. Rudman, An sph projection method, *J. Comput. Phys.* 152 (2) (1999) 584–607.
- [60] R. Xu, P. Stansby, D. Laurence, Accuracy and stability in incompressible sph (isph) based on the projection method and a new approach, *J. Comput. Phys.* 228 (18) (2009) 6703–6725.
- [61] G. Fourtakas, P. Stansby, B.D. Rogers, S. Lind, An Eulerian–Lagrangian incompressible sph formulation (eli-sph) connected with a sharp interface, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 532–552.
- [62] F.M. White, *J. Majdalan, Viscous Fluid Flow*, vol. 3, McGraw-Hill, New York, 2006.
- [63] P.-H. Chiu, R.-K. Lin, T.W. Sheu, A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries, *J. Comput. Phys.* 229 (12) (2010) 4476–4500.
- [64] D.-V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.* 220 (1) (2006) 109–138.
- [65] C. Brehm, C. Hader, H.F. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 295 (2015) 475–504.
- [66] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2d incompressible viscous flow, *J. Comput. Phys.* 191 (1) (2003) 177–205.

- [67] L. Qu, C. Norberg, L. Davidson, S.-H. Peng, F. Wang, Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200, *J. Fluids Struct.* 39 (2013) 347–370.
- [68] G.F. Acosta, J. Calderon-Sanchez, P.E. Merino-Alonso, R. Zamora-Rodriguez, A hydrodynamic study of various obstacle shapes in 2d flow using sph, *Ocean Eng.* 310 (2024) 118722.
- [69] C. Huang, T. Long, S. Li, M. Liu, A kernel gradient-free sph method with iterative particle shifting technology for modeling low-Reynolds flows around airfoils, *Eng. Anal. Bound. Elem.* 106 (2019) 571–587.
- [70] H.C. Ku, R.S. Hirsh, T.D. Taylor, A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations, *J. Comput. Phys.* 70 (2) (1987) 439–462.
- [71] D. Komaizi, A. Niknam, Hybrid finite-volume–finite-element scheme for 3d simulation of thermal plasma arc configuration, *AIP Adv.* 14 (1) (2024).
- [72] K. Kitamura, E. Shima, Simple and parameter-free second slope limiter for unstructured grid aerodynamic simulations, *AIAA J.* 50 (6) (2012) 1415–1426.
- [73] K. Iwasaki, S.-i. Inutsuka, Smoothed particle magnetohydrodynamics with a Riemann solver and the method of characteristics, *Mon. Not. R. Astron. Soc.* 418 (3) (2011) 1668–1688.

A.2 Paper II

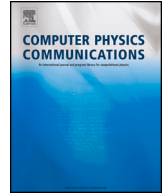
Zhentong Wang, Oskar J. Haidn, Xiangyu Hu

An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework

In *Computer Physics Communications*, Volume 307, 2025, p. 109429, DOI: <https://doi.org/10.1016/j.cpc.2024.109429>.

Copyright © 2025 Elsevier. Reprinted with permission.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.



Computational Physics

An algorithm for the incorporation of relevant FVM boundary conditions in the Eulerian SPH framework[☆]Zhentong Wang, Oskar J. Haidn, Xiangyu Hu^{*}

TUM School of Engineering and Design, Technical University of Munich, Garching, 85748, Germany

ARTICLE INFO

Keywords:

Boundary algorithm
Finite volume method
Smoothed particle hydrodynamics
Strong coupling
Zero-order consistency
SPHInXsys

ABSTRACT

The finite volume method (FVM) is widely recognized as a computationally efficient and accurate mesh-based technique. However, it has notable limitations, particularly in mesh generation and handling complex boundary interfaces or conditions. In contrast, the smoothed particle hydrodynamics (SPH) method, a popular meshless alternative, inherently circumvents the challenges of mesh generation and yields smoother numerical outcomes. Nevertheless, this approach comes at the cost of reduced computational efficiency. Consequently, researchers have strategically combined the strengths of both methods to investigate complex flow phenomena, producing precise and computationally efficient outcomes. However, algorithms involving the weak coupling of these two methods tend to be intricate and face challenges regarding versatility, implementation, and mutual adaptation to hardware and coding structures. Thus, achieving a robust and strong coupling of FVM and SPH within a unified framework is essential. A mesh-based FVM has recently been integrated into the SPH-based library SPHInXsys. However, due to the differing boundary algorithms between these methods, the crucial step for establishing a strong coupling of both methods within a unified SPH framework is to incorporate the FVM boundary algorithm into the Eulerian SPH method. In this paper, we propose a straightforward algorithm within the Eulerian SPH method, which is algorithmically equivalent to that in FVM and based on the principle of zero-order consistency. Moreover, several numerical examples, including compressible and incompressible flows with various boundary conditions in the Eulerian SPH method, demonstrate the stability and accuracy of the proposed algorithm.

1. Introduction

In the context of rapidly advancing high-performance computing, computational fluid dynamics (CFD) has emerged as a highly effective approach in addressing industrial challenges and has proven instrumental in elucidating longstanding flow phenomena across a wide range of scales, ranging from microfluidics to hydrodynamics and hypersonics [1,2]. While traditional mesh-based CFD methods have achieved significant accomplishments, they face limitations when addressing complex practical problems. These challenges pertain to generating high-quality meshes and modeling intricate interfaces, such as solid-fluid moving boundaries [3] and violent deformations (e.g., wave breaking). Fortunately, to address the latter constraint, traditional mesh-based techniques, such as the finite volume method (FVM), can be coupled with other numerical techniques, including both mesh-based [4–7] and meshless [8–13] methods, to achieve convincing and accurate results. Nevertheless, the former challenge is inherent to complex geometric settings

in mesh-based methods. By avoiding mesh generation and accommodating large deformations during simulations, meshless methods have attracted significant interest since their numerical formulation is based on particles and independent of a mesh-defined topology. Among the typical meshless methods, the smoothed particle hydrodynamics (SPH) method is characterized by numerical approximations using Gaussian-like kernel functions [14,15] and has found widespread application in diverse fields including CFD [16–18], structural mechanics [19], and various scientific and engineering domains [20,21], particularly in scenarios where traditional mesh-based methods encounter challenges.

As previously indicated, numerous researchers have combined FVM with meshless approaches to tackle intricate flows characterized by complex interfaces or boundaries. Similarly, they have also combined SPH with traditional mesh-based methods to improve computational efficiency and address complex boundary conditions [9,10,22]. Consequently, the fusion of mesh-based methods such as FVM and meshless methods such as SPH is crucial for solving complex fluid dynamics prob-

[☆] The review of this paper was arranged by Prof. Andrew Hazel.^{*} Corresponding author.E-mail addresses: zhentong.wang@tum.de (Z. Wang), oskar.haidn@tum.de (O.J. Haidn), xiangyu.hu@tum.de (X. Hu).

lems. Nevertheless, it is recognized that the weak coupling of these multiple methods introduces inherent limitations. Firstly, there is the potential for reduced numerical stability when combining two distinct methods, compared to using a single method, due to the need for tailored algorithms to manage data transfer or exchange within overlapping regions or interfaces [8,10,23,24]. Secondly, the implementation and generalization of coupling algorithms present significant challenges [25], primarily due to the inherent divergence in the fundamental characteristics of the coupled models. Thirdly, apart from variations in formulations, another notable challenge arises when attempting to couple methodologies that are designed or highly optimized for distinct types of hardware acceleration and coding constructs [25]. Hence, introducing a unified framework for computing complicated CFD problems becomes essential to address these limitations. Wang et al. [26] have proposed a certain form of the Eulerian SPH method that is equivalent to FVM and have implemented the mesh-based FVM within the open-source SPHinxsys [27,28], paving the way for the strong coupling of these two methods within a unified SPH framework. While both methods in Wang et al. [26] share the same discretized conservation equations and other techniques for improving stability and accuracy, it is essential to note that the treatment of boundary conditions differs between the particle-based and mesh-based approaches. Based on this, the incorporation of the FVM boundary algorithm into the Eulerian SPH method becomes a critical issue demanding a solution, as it not only enhances the equivalence between the Eulerian SPH and FVM but also constitutes a crucial component in establishing a unified SPH framework for achieving the robust and strong coupling of these two methods.

In this paper, by analyzing the boundary algorithm in FVM, which ensures that each cell in the boundaries adheres to zero-order consistency, we propose a new boundary algorithm adapted to the Eulerian SPH framework. While implementing the boundary algorithm in FVM within SPHinxsys, we allocate additional storage space for particles in the boundaries to ensure that the necessary data is stored to meet the requirements of FVM's boundary treatment. Additionally, a series of numerical examples with various boundary conditions, including reflecting, inflow, outflow, moving, non-reflecting, non-slip wall, and far-field boundary conditions, are tested in the Eulerian SPH framework to validate the stability and accuracy of the proposed algorithm.

The structure of this article is as follows: Section 2 presents the governing equations and their discretization, together with a low-dissipation limiter proposed by Zhang et al. [29] for a Riemann-based SPH scheme [30,31] to reduce numerical dissipation, and a kernel correction method [32,33] for the gradient operator to further enhance numerical accuracy. This section also discusses a particle relaxation method developed by Zhu et al. [34], which uses a background pressure to physically move and generate body-fitted particles while maintaining zero-order consistency. Additionally, Section 3 details the implementation of the new boundary algorithm in the Eulerian SPH method through a comparison with the corresponding approach in FVM. Finally, Section 4 evaluates the performance of the proposed algorithm through several examples with various boundary conditions. The computational codes used in this study are readily available through the SPHinxsys repository and can be accessed at <https://www.sphinxsys.org> or <https://github.com/Xiangyu-Hu/SPHinxsys>.

2. Governing equations and Eulerian SPH method

2.1. Governing equations

The conservation equations within the Eulerian framework can be described as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (1)$$

where $\mathbf{F}(\mathbf{U})$ represents the fluxes corresponding to $\mathbf{U}$, which denotes the vector of conserved variables. In a two-dimensional case, these are expressed as:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}, \quad (2)$$

where u and v are the components of velocity, ρ and p denote the density and pressure, respectively, and $E = \frac{\rho(u^2 + v^2)}{2} + \rho e$ is the total energy, where e represents the internal energy. The equation of state (EOS) is introduced to close Eq. (1) and is given by:

$$p = \begin{cases} \rho(\gamma - 1)e & \text{For compressible flows} \\ c^2(\rho - \rho_0) & \text{For incompressible flows} \end{cases}. \quad (3)$$

Here, γ represents the heat capacity ratio, ρ_0 and c are the reference density and speed of sound, respectively. For compressible flows, the speed of sound is derived from the ideal gas equation as follows:

$$c^2 = \frac{\gamma p}{\rho}. \quad (4)$$

In the case of incompressible flows, the weakly compressible assumption is adopted by employing an artificial EOS expressed in Eq. (3). To restrict density variations to within 1%, the speed of sound is defined as $c = 10U_{max}$, where U_{max} is the anticipated maximum velocity within the flow field. Notably, in this weakly compressible formulation, the energy conservation equation in Eq. (2) is disabled. To simulate viscous flows, a viscous force is incorporated into the momentum equation and is defined as:

$$f^{(v)} = \eta \nabla^2 \mathbf{v}, \quad (5)$$

where η and $\mathbf{v}$ denote the dynamic viscosity and fluid velocity, respectively.

2.2. Low-dissipation Riemann-based Eulerian SPH discretization

Following Vila [30], the discretized form of Eq. (1) can be expressed as follows:

$$\begin{cases} \frac{\partial}{\partial t} (w_i \rho_i) + 2w_i \sum_j w_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (w_i \rho_i \mathbf{v}_i) + 2w_i \sum_j w_j \left[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbb{I} \right] \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (w_i E_i) + 2w_i \sum_j w_j \left[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^* \right] \cdot \nabla W_{ij} = 0 \end{cases}. \quad (6)$$

In this formulation, w and $\mathbb{I}$ denote the particle volume and the identity matrix, respectively. Here, the gradient of the kernel function, $\nabla W_{ij} = \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}} \mathbf{e}_{ij}$, is represented with the unit vector $\mathbf{e}_{ij} = -\mathbf{r}_{ij}/r_{ij}$, where $\mathbf{r}_{ij}$ is the displacement vector from particle j to particle i , and r_{ij} denotes the corresponding distance $|\mathbf{r}_{ij}|$. Note that $|\mathbf{r}_{ij}|$ can also represent the distance between the particles in the boundaries and the physical boundary surface, which will be detailed in Section 3. Moreover, the terms denoted by $(\cdot)_{E,ij}^*$ correspond to the solutions of the Riemann problem [30]. For viscous flows, the viscous term is discretized as follows:

$$f_i^{(v)} = 2 \sum_j w_j \frac{\eta}{\rho_i} \frac{\mathbf{v}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}}, \quad (7)$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ represents the velocity difference between particles i and j .

To solve the Riemann problem, where the initial left and right states are on particles i and j , respectively, and the l and r states are given by:

$$\begin{cases} (\rho_l, u_l, p_l, E_l) = (\rho_i, \mathbf{v}_i \cdot \mathbf{e}_{ij}, p_i, E_i) \\ (\rho_r, u_r, p_r, E_r) = (\rho_j, \mathbf{v}_j \cdot \mathbf{e}_{ij}, p_j, E_j) \end{cases}, \quad (8)$$

three waves with varying speeds are employed: the smallest denoted as S_l , the middle as S_* , and the largest as S_r . Note that the middle wave distinguishes between the two intermediate states, defined by $(\rho_l^*, u_l^*, p_l^*, E_l^*)$ and $(\rho_r^*, u_r^*, p_r^*, E_r^*)$. For compressible flows, the HLLC Riemann solver [35,36] is employed due to its effectiveness in capturing

shock discontinuities. The estimation of the smallest and largest wave speeds is given by:

$$S_l = \bar{u} - \bar{c}, S_r = \bar{u} + \bar{c}, \quad (9)$$

where $\bar{u}$ and $\bar{c}$ denote the density-average particle and sound speed, respectively, and are calculated by:

$$\begin{cases} H_l = \frac{E_l + p_l}{\rho_l}, H_r = \frac{E_r + p_r}{\rho_r}, \tilde{H} = \frac{\rho_l H_l + \rho_r H_r}{\rho_l + \rho_r} \\ \bar{u} = \frac{\rho_l u_l + \rho_r u_r}{\rho_l + \rho_r}, \bar{c} = \sqrt{(\gamma - 1)(\tilde{H} - 0.5\bar{u}^2)} \end{cases}, \quad (10)$$

where H denotes the enthalpy and $\tilde{H}$ the density-weighted average enthalpy. The middle wave speed is further obtained using:

$$S_* = \frac{\rho_r u_r (S_r - u_r) + \rho_l u_l (u_l - S_l) + p_l - p_r}{\rho_r (S_r - u_r) + \rho_l (u_l - S_l)}. \quad (11)$$

Subsequently, the remaining states within the star region can be deduced as follows:

$$p^* = p_l + \rho_l (u_l - S_l) (u_l - u^*) = p_r + \rho_r (S_r - u_r) (u^* - u_r), \quad (12)$$

$$\mathbf{v}^* = u^* \mathbf{e}_{ij} + \left[\frac{1}{2}(\mathbf{v}_l + \mathbf{v}_r) - \frac{1}{2}(u_l + u_r) \mathbf{e}_{ij} \right], \quad (13)$$

$$\rho_{l/r}^* = \rho_{l/r} \frac{(S_{l/r} - u_{l/r})}{(S_{l/r} - u^*)}, \quad (14)$$

$$E_{l/r}^* = \frac{(S_{l/r} - u_{l/r}) E_{l/r} - p_{l/r} u_{l/r} + p^* u^*}{S_{l/r} - u^*}. \quad (15)$$

Here, $u^* = S_*$. For incompressible fluid flows, assuming that the intermediate states satisfy $p_l^* = p_r^* = p^*$ and $u_l^* = u_r^* = u^*$, a linearized Riemann solver [37] can be expressed as

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (16)$$

where $\bar{\rho}$ and $\bar{c}$ represent density-weighted averages.

To decrease the numerical dissipation in the Riemann problem, some high-order methods, such as second-order Monotone Upwind-centered Scheme for Conservation Laws (MUSCL) [38], weighted essentially non-oscillatory (WENO) [39] and third-order Piecewise Parabolic Method (PPM) [40], have been developed [20]. However, these methods significantly increase the computational load due to the need for time-consuming data reconstructions. To maintain computational efficiency and enhance numerical accuracy, a low-dissipation limiter in the HLLC Riemann solver for compressible flows is introduced [26], and the middle wave speed and pressure in the star region can be given by

$$\begin{cases} u^* = \frac{\rho_l u_l c_l + \rho_r u_r c_r}{\rho_l c_l + \rho_r c_r} + \frac{p_l - p_r}{\rho_l c_l + \rho_r c_r} \beta_{HLLC}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{HLLC} [\rho_r c_r (u^* - u_r) - \rho_l c_l (u_l - u^*)] \end{cases}. \quad (17)$$

Here, the limiter [26] is defined as:

$$\beta_{HLLC} = \min \left(\eta_{HLLC} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (18)$$

Note that there is no dissipation when the fluid undergoes an expansion wave, i.e., when $u_l < u_r$, whereas the parameter η_{HLLC} modulates dissipation during a compression wave, i.e., when $u_l > u_r$. Here, $\eta_{HLLC} = 1$ is suggested according to the numerical experiment to ensure overall numerical stability and accuracy for compressible flow cases within the Eulerian SPH framework.

As for the incompressible flows, the middle wave speed and pressure in the linearised Riemann solver can be calculated as

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \beta_{linearisd}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{linearisd} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (19)$$

where the limiter is defined as

$$\beta_{linearisd} = \min \left(\eta_{linearisd} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (20)$$

Here, $\eta_{linearisd} = 15$ is recommended based on numerical experiments and overall numerical stability and accuracy for incompressible flow cases. Note that, unlike the limiter used in the Lagrangian SPH framework [29], the limiter in the velocity term is also introduced within the Eulerian SPH framework. This addition aims to reduce the extra numerical dissipation arising from the convection term in the momentum equation. Moreover, the value of $\eta_{HLLC} = 1$ in compressible flows differs from the value of $\eta_{linearisd} = 15$ in incompressible flows, as the compressible framework induces additional numerical dissipation in the energy equation compared to the incompressible framework. To further improve numerical accuracy, a kernel gradient correction [26,32,33] is also introduced:

$$\mathbf{B}_i = - \left(\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} w_j \right)^{-1}. \quad (21)$$

To maintain the conservative property, the kernel correction in a conservative form is defined as:

$$\nabla' W_{ij} = \frac{\mathbf{B}_i + \mathbf{B}_j}{2} \nabla W_{ij}, \quad (22)$$

to replace the original kernel gradient ∇W_{ij} in Eq. (6). Details regarding implementing the kernel correction will be explained in Section 3.

2.3. Particle relaxation technique

It is widely acknowledged that particle distribution significantly influences numerical accuracy [41]. In the SPH method, the initial particle setup is typically a lattice distribution, which in practical applications often fails to accurately fit the surface of complex shapes. One approach to overcome this limitation is to use very fine spatial resolution [42], which results in significant computational demands. Although some methods [43–45] have been developed to achieve optimal particle distribution, they still face challenges with highly complex geometries owing to either algorithmic complexity or increased computational burden. This paper adopts a straightforward and effective method proposed by Zhu et al. [34] to generate the body-fitted particle distribution.

In this method, the geometry is imported before particle relaxation. The initial particles with lattice distribution are physically moved under a constant background pressure, expressed as

$$\mathbf{a}_i = - \frac{2}{\rho_i} \sum_j w_j \nabla W_{ij}. \quad (23)$$

When the acceleration $\mathbf{a}_i$ reaches zero, the particles not in the boundaries attain stable positions and approximately satisfy the condition $\sum_j w_j \nabla W_{ij} \approx \mathbf{0}$, with errors kept within a small percentage range. Consequently, the relaxed particle distribution is considered to have achieved zero-order consistency of $\sum_j w_j \nabla W_{ij} = \mathbf{0}$, i.e., reproducing the constant field. Furthermore, the impact of kernel correction on the zero-order consistency of particles not in the boundaries within the relaxed particle distribution is investigated. A quantitative analysis is conducted on a circular geometry with a diameter of $D = 2$, using a resolution of $dp = D/20$, to perform the L_1 -norm error analysis of the SPH approximation of df/dx with a constant function $f = 1$, both with and without kernel correction. Here, the L_1 -norm error is defined by $L_1(df/dx) = \frac{1}{N} \sum_{i=1}^N |(\nabla f_i)^{Theoretical} - (\nabla f_i)^{Numerical}|$, where $(\nabla f_i)^{Theoretical}$ and $(\nabla f_i)^{Numerical}$ denote the analytical and SPH approximated results, respectively. The L_1 -norm error is 2.9693×10^{-4} without the kernel correction and 2.9636×10^{-4} with the kernel correction. This indicates that the kernel correction exerts minimal influence on zero-order consistency within the relaxed particle distribution, thereby maintaining the zero-order consistency of particles not in the boundaries. Note that particles in the boundaries do not meet the zero-order

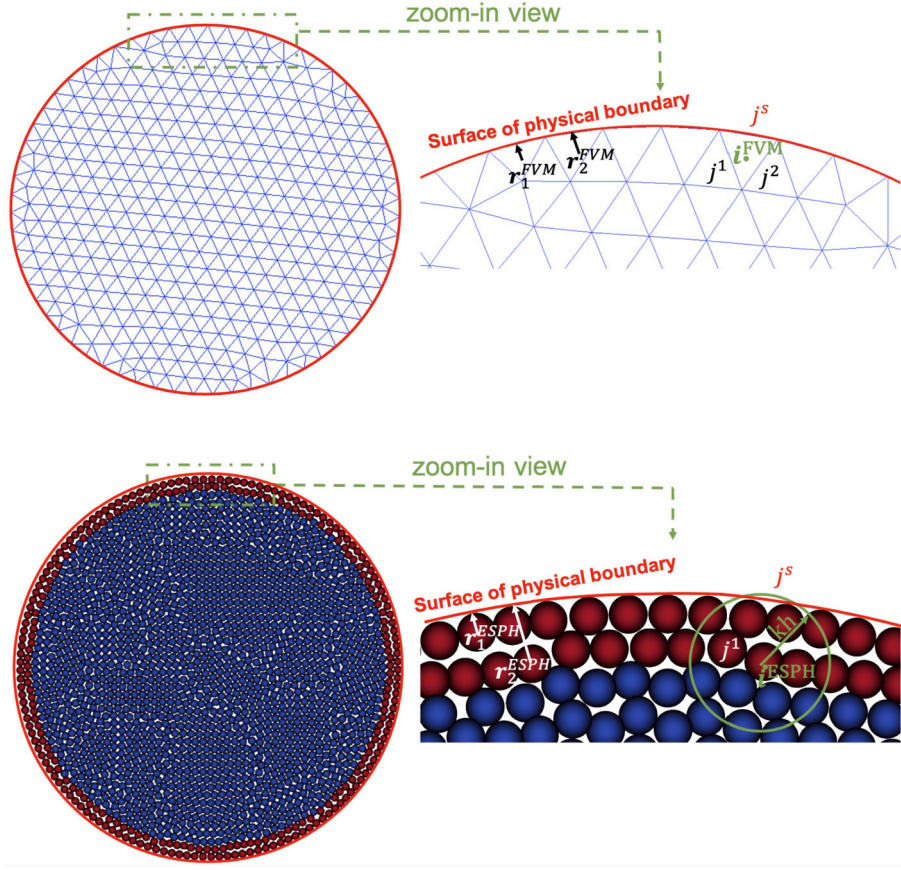


Fig. 1. Mesh (top panel) and particle (bottom panel) distributions and their zoom-in views in FVM and Eulerian SPH, respectively, for a circular geometry. Note that the red circular line is the surface of the geometry. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

consistency criteria due to the lack of neighboring particles within the compact support. The solution to this issue will be elaborated in the forthcoming Section 3. Besides, following khayyer et al. [46], the error $L_1(\rho) = \frac{1}{N} \sum_{i=1}^N |\hat{\rho}_i - \rho_0|$, where $\hat{\rho}_i = \sum_j w_j W_{ij}$ represents the normalized density [46], is further analyzed to assess the normalization condition property, i.e., $\sum_j w_j W_{ij} = 1$. The error is approximately 0.28%, within the allowable range of 1% [46], confirming that the normalization condition is well-maintained.

3. Boundary algorithm in Eulerian SPH framework

In this section, we elaborate the boundary algorithm in Eulerian SPH framework through a comparison with boundary properties in the mesh-based FVM.

The discretized conservation Eq. (6) in both methods [26,47] is expressed as follows:

$$\frac{\partial}{\partial t} (\omega_i \mathbf{U}_i) + \sum_j \mathbf{F}_{ij}(\mathbf{U}) \cdot \mathbf{A}_{ij} = 0, \quad (24)$$

with $\mathbf{A}_{ij}$ defined as:

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{A}_{ij}^{ESPH} = |\mathbf{A}_{ij}| \mathbf{n}_{ij}^{ESPH} = 2\omega_i \omega_j \nabla W_{ij} & \text{In Eulerian SPH} \\ \mathbf{A}_{ij}^{FVM} = |\mathbf{A}_{ij}| \mathbf{n}_{ij}^{FVM} = S_{ij} \mathbf{n}_{ij}^{FVM} & \text{In FVM} \end{cases}, \quad (25)$$

where $|\mathbf{A}_{ij}|$ represents the interface size, with S_{ij} specifically denoting the interface size between cell i and j in FVM. Here, $\mathbf{n}_{ij}^{ESPH}$ and $\mathbf{n}_{ij}^{FVM}$ are the unit normal vectors at the interface in Eulerian SPH and FVM, respectively. For clarity, in the following content, we categorize all cells in FVM and all particles in Eulerian SPH into two groups: those in the

boundaries and those not in the boundaries. For the former, we further differentiate the flux contributions from two components: neighboring cells or particles (labeled as j^N) and the boundary surface (labeled as j^S). For the latter, j^N represents all neighboring particles within the whole compact support or all neighboring cells.

In FVM, cells not in the boundaries form completely closed control volumes [26,47], naturally and strictly adhering to the condition $\sum_j \mathbf{A}_{ij}^{FVM} = \mathbf{0}$, i.e., the zero-order consistency. However, for cells in the boundaries, i.e., connecting the boundary surface, such as cell i^{FVM} in Fig. 1 (top panel), the control volume must be enclosed by the boundary surface. Therefore, all cells strictly satisfy the zero-order consistency. To enhance understanding, the zero-order consistency of the cells in the boundaries can be rewritten as

$$\sum_j \mathbf{A}_{ij}^{FVM} = \sum_{j^N} \mathbf{A}_{ij^N}^{FVM} + \mathbf{A}_{ij^S}^{FVM} = \mathbf{0}. \quad (26)$$

Here, in a two-dimensional unstructured mesh where $N = 1$ or 2 , as shown by j^1 and j^2 in Fig. 1 (top panel), $\sum_{j^N} \mathbf{A}_{ij^N}^{FVM} = \sum_{j^N} S_{ij^N} \mathbf{n}_{ij^N}^{FVM}$ determines the flux contribution from all neighboring cells. Similarly, $\mathbf{A}_{ij^S}^{FVM} = S_{ij^S} \mathbf{n}_{ij^S}^{FVM}$ determines the flux contribution from the boundary surface. Note that $\mathbf{A}_{ij^S}^{FVM}$ can be calculated directly within the framework of FVM, as the interface size S_{ij^S} and the unit normal vector $\mathbf{n}_{ij^S}^{FVM}$ can be obtained from the mesh file.

In Eulerian SPH, the SPH approximation relies on identifying neighboring particles within the compact support [14]. Once the particle relaxation process in Eq. (23) is completed, particles not in the boundaries can be assumed to satisfy zero-order consistency. However, particles in the boundaries, like particle i^{ESPH} as depicted in Fig. 1 (bottom panel), lack neighboring particles and thus fail to meet the zero-order consistency.

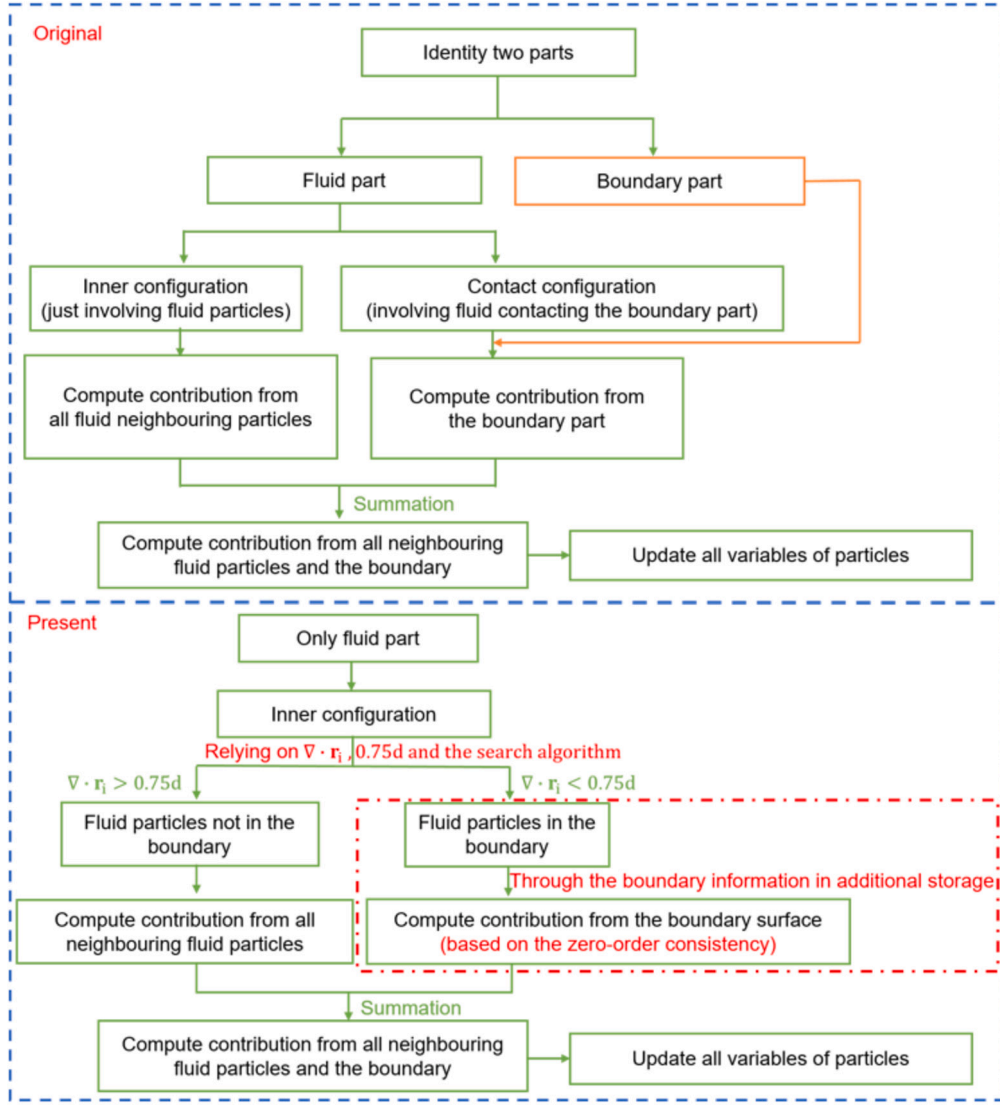


Fig. 2. Flowcharts of the configurations in Eulerian SPH method with the original algorithm labeled as “Original” in the top panel and the proposed algorithm labeled as “Present” in the bottom panel are presented in the case of fluid flows with designed boundary conditions. These flowcharts depict the process of computing contributions and updating variables based on the calculated contributions at the new time step.

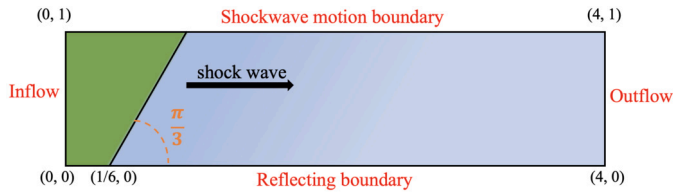


Fig. 3. Double Mach reflection of a strong shock: The geometry and boundary conditions.

tency criteria. To address this issue, similar to Eq. (26), we introduce the strict zero-order consistency in Eulerian SPH, expressed as

$$\sum_j \mathbf{A}_{ij}^{ESPH} = \sum_{jN} \mathbf{A}_{ijN}^{ESPH} + \mathbf{A}_{ijS}^{ESPH} = \mathbf{0}, \quad (27)$$

where $\sum_{jN} \mathbf{A}_{ijN}^{ESPH} = 2\omega_i \omega_{jN} \sum_{jN} \nabla W_{ijN}$ determines the flux contribution from all neighboring particles. Additionally, $\mathbf{A}_{ijS}^{ESPH}$ determines the flux contribution from the boundary surface. Note that, unlike $\mathbf{A}_{ijS}^{FVM}$ in FVM, $\mathbf{A}_{ijS}^{ESPH}$ cannot be calculated directly as the boundary surface

size cannot be easily obtained in Eulerian SPH. Thus, $\mathbf{A}_{ijS}^{ESPH}$ can be calculated from Eq. (27) as follows:

$$\mathbf{A}_{ijS}^{ESPH} = - \sum_{jN} \mathbf{A}_{ijN}^{ESPH}. \quad (28)$$

Note that Eq. (28) is applied twice in the present boundary algorithm as follows: Firstly, after completing the particle relaxation process and achieving a relaxed particle distribution, regardless of whether it is a Cartesian or non-Cartesian distribution, particles not in the boundaries approximately satisfy zero-order consistency. However, particles in the boundaries do not satisfy this consistency. At this stage, Eq. (28) is applied for the first time to obtain $\mathbf{A}_{ijS}^{ESPH}$, ensuring these particles in the boundaries adhere to zero-order consistency. Secondly, to enhance numerical accuracy, the kernel correction is implemented within the relaxed particle distribution, updating the kernel gradient of Eq. (22). Consequently, $\sum_{jN} \mathbf{A}_{ijN}^{ESPH}$ is updated, and this update can potentially cause Eq. (28) to deviate from strict satisfaction. To address this, Eq. (28) is applied a second time to recalculate $\mathbf{A}_{ijS}^{ESPH}$ using the updated $\sum_{jN} \mathbf{A}_{ijN}^{ESPH}$. Thus, particles not in the boundaries achieve approximate zero-order consistency with minimal error, as discussed in Section 2.3,

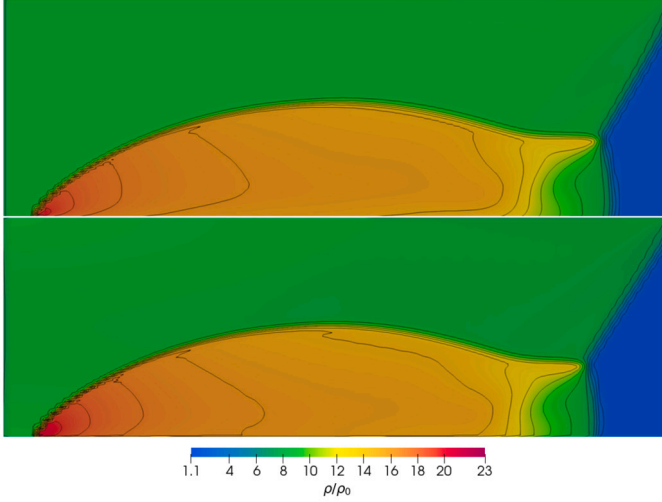


Fig. 4. Double Mach reflection of a strong shock: 19 equally spaced density (ρ/ρ_0) contour obtained by the original (top panel) and present (bottom panel) boundary algorithms ranging from 1.1 to 23 with the spatial resolution $dp/L_0 = 1/128$ in the Eulerian SPH method at the time $tU_0/L_0 = 0.2$.

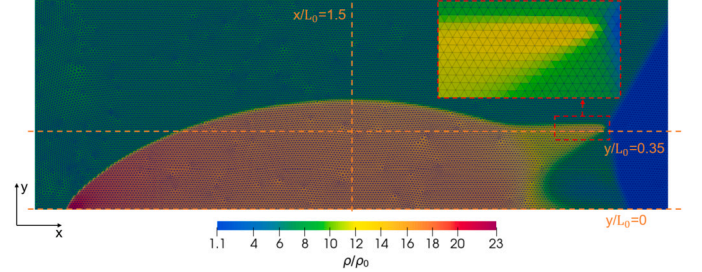


Fig. 5. Double Mach reflection of a strong shock: The density (ρ/ρ_0) contour, ranging from 1.1 to 23, obtained by FVM with the unstructured mesh consisting of a total cell number $N = 64162$, along with a zoom-in view for clearer visualization of the unstructured mesh distribution, at the time $tU_0/L_0 = 0.2$.

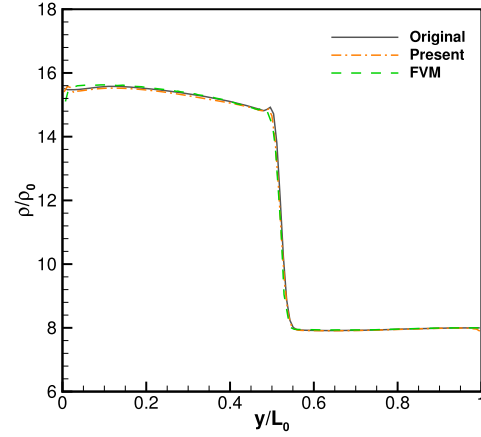
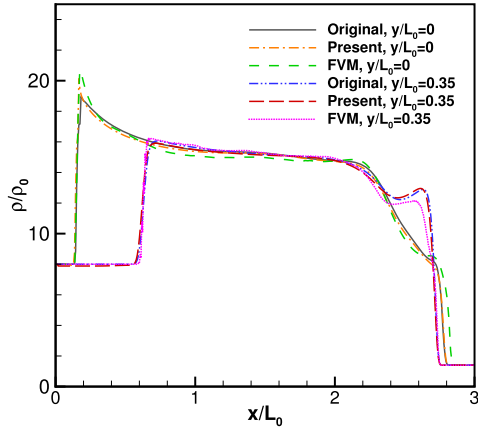


Fig. 6. Double Mach reflection of a strong shock: The density (ρ/ρ_0) profiles obtained using the original and present boundary algorithms, and FVM methods along $y/L_0 = 0$, $y/L_0 = 0.35$, and $x/L_0 = 1.5$.

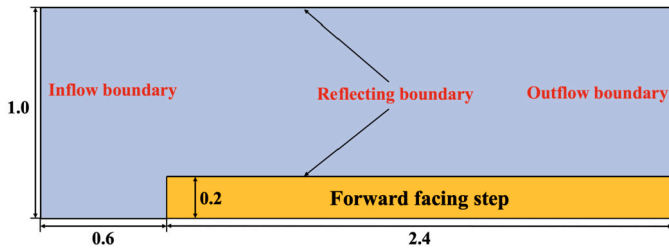


Fig. 7. A Mach 3 wind tunnel with a step: The geometry and boundary conditions.

while particles in the boundaries strictly adhere to zero-order consistency. With this consistency satisfied, the flux contribution from the boundary surface is obtained accurately, thus avoiding instabilities like the one-sided flux problem.

Having established the algorithm for particles in the boundaries in Eulerian SPH, we now proceed with its implementation. The initial step is to accurately identify and label the particles in the boundaries. To accomplish this, we introduce the divergence of position [48,49], expressed as:

$$\nabla \cdot \mathbf{r}_i = \sum_j w_j \mathbf{r}_{ij} \cdot \nabla W_{ij}, \quad (29)$$

where j represents all neighboring particles, i.e., $j = j^N$. Fluid particles within the full kernel support size satisfy $\nabla \cdot \mathbf{r}_i \approx d$, with d denoting the dimension, and are classified using Eq. (29) and a threshold value of $\gamma = 0.75d$ as either forming or not forming the surface. Particles are categorized as forming the surface if their position divergence is below this threshold value and subsequently labeled as “red”. Most “red” particles identified using Eq. (29) and the threshold value are located in the first outer layer along the boundary surface. However, relying exclusively on Eq. (29) is insufficient and leads to the failure to detect certain particles, especially in corner regions and complex geometries. To address this limitation, we implement a search algorithm proposed by Zhang et al. [49]. In this approach, all particles, except those already labeled as “red”, search for their neighboring particles within the cutoff radius of kh with $k = 1$, as illustrated by the green circle in Fig. 1 (bottom panel). If any neighboring particles are labeled as “red”, these particles are also labeled as “red”. Based on this criterion, particles in the outer two layers, colored by red particles in Fig. 1 (bottom panel), are identified as particles in the boundaries. It is worth noting that the surface of the physical boundary is represented consistently in both FVM and Eulerian SPH frameworks, as depicted by the red circular line in Fig. 1, where

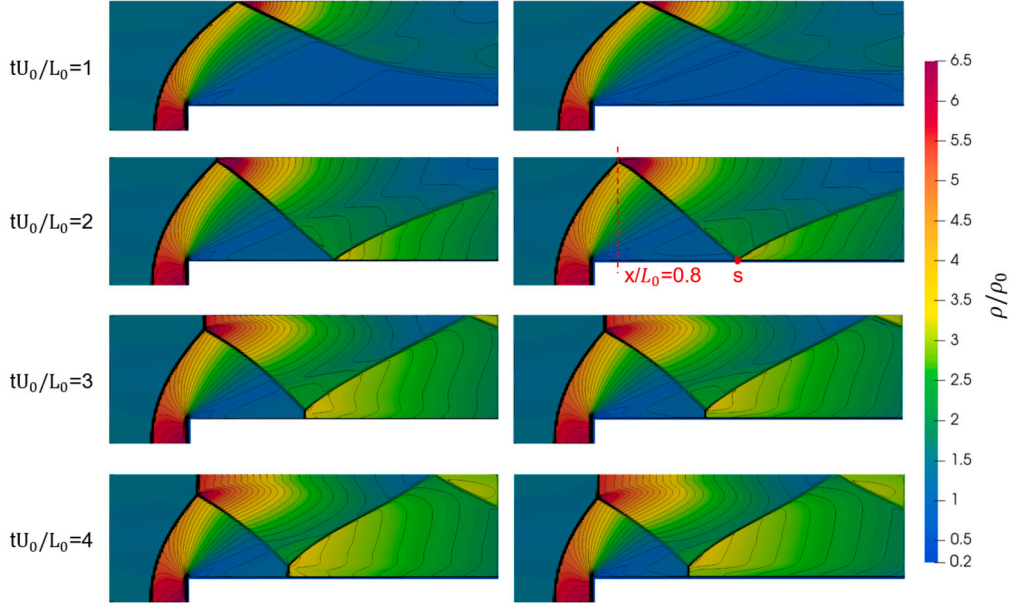


Fig. 8. A Mach 3 wind tunnel with a step: 30 equally spaced density (ρ/ρ_0) contour ranging from 0.2 to 6.5 obtained by the original (left panel) and present (right) boundary algorithms with the spatial resolution $dp/L_0 = 1/200$ at the different time instants including $tU_0/L_0 = 1, 2, 3$ and 4.

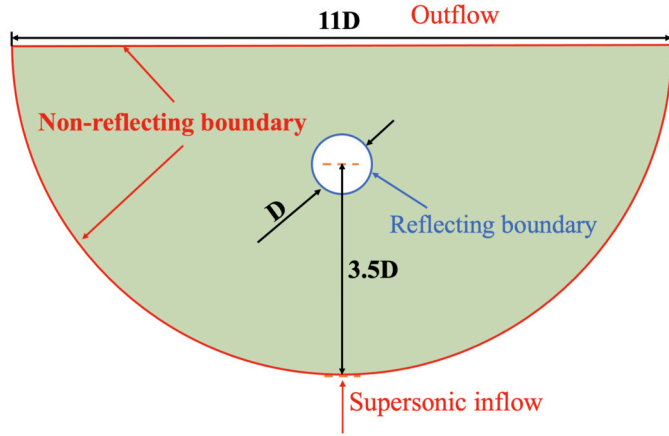


Fig. 9. Supersonic flow past a circular cylinder: The geometry and boundary conditions.

$\mathbf{r}^{ESPH}$ and $\mathbf{r}^{FVM}$ represent the displacement from particles and cells in the boundaries to the boundary surface in the Eulerian SPH and FVM, respectively. One difference is that while, in FVM, the cells having the flux contribution from the boundaries are those that connect directly to the surface of the physical boundary, the particles in Eulerian SPH having the flux contribution from the boundaries are those labeled as “red”, i.e., two outer layer particles.

The second step involves the computation of $\mathbf{A}_{ijS}^{ESPH}$ for the particles in the boundaries. In the original boundary approach [50,26,51] within the SPH framework, boundary conditions, such as the wall boundary condition [50], are typically applied to predefined particles within the contact configurations, in addition to the fluid particles within the inner configuration. Then, the total flux is calculated from both inner and contact configurations to update all particle variables at each new time step. In contrast, the present algorithm requires only the inner configuration. Fig. 2 provides flowchart diagrams detailing the configurations and the computation of flux contributions for updating particle variables using the original and present algorithms in Eulerian SPH method. In the present algorithm, $\sum_{jN} \mathbf{A}_{ijN}^{ESPH}$ can be calculated after the particles in the boundaries are identified in the initial step. Subsequently, $\mathbf{A}_{ijS}^{ESPH}$

can be obtained from Eq. (28). Finally, after $\sum_{jN} \mathbf{A}_{ijN}^{ESPH}$ is updated with introducing the kernel gradient correction, $\mathbf{A}_{ijS}^{ESPH}$ is recalculated using Eq. (28), as highlighted in the red dashed square in Fig. 2.

The third step involves allocating additional data storage space for the particles in the boundaries. This storage is designed to retain essential information, including the magnitude of flux contribution from the boundary surface, $|\mathbf{A}_{ijS}^{ESPH}|$, the normal unit vector of the flux contribution, $\frac{\mathbf{A}_{ijS}^{ESPH}}{|\mathbf{A}_{ijS}^{ESPH}|}$, the distance between the particle and the physical

boundary surface, $|\mathbf{r}^{ESPH}|$ such as $|\mathbf{r}_1^{ESPH}|$ and $|\mathbf{r}_2^{ESPH}|$ in Fig. 1 (bottom panel), and the specified boundary condition values. This additional storage is employed to calculate the flux from the boundary surface for updating particle variables. In SPHinXsys, between a pairwise particle i and j , the kernel gradient ∇W_{ij} multiplying the neighboring particle volume, meaning the flux contribution from the particle j to particle i , is stored separately as the magnitude of $\frac{\partial W_{ij}}{\partial r_{ij}} w_j$ and the displacement unit vector of $\mathbf{e}_{ij}$. Based on this data structure, the flux contribution from the boundary surface, $\mathbf{A}_{ijS}^{ESPH}$, is separately stored as the magnitude $|\mathbf{A}_{ijS}^{ESPH}|$ in data storage of $\frac{\partial W_{ij}}{\partial r_{ij}} w_j$ and normal unit vector $\frac{\mathbf{A}_{ijS}^{ESPH}}{|\mathbf{A}_{ijS}^{ESPH}|}$

in data storage of $\mathbf{e}_{ij}$. Additionally, the distance between these particles and the boundary surface $|\mathbf{r}^{ESPH}|$, which is utilized to calculate the viscous force from the boundary surface in Eq. (7), can be easily obtained in SPHinXsys and is stored in the data storage of $|\mathbf{r}_{ij}|$.

The fourth step entails the implementation of boundary conditions. The proposed algorithm is compatible with various boundary conditions, such as reflecting, inflow, outflow, moving, non-reflecting, non-slip wall, and far-field boundaries. The designated values are input into the additional storage for each boundary condition. For instance, if a time-independent inflow boundary condition with a given constant density and velocity is applied, these values are retained in the additional storage for density and velocity throughout the simulation. For time-dependent boundary conditions with variable values, such as a rigid wall boundary condition (following the one-sided Riemann problem [50]), the density, pressure, and velocity in the additional storage are adjusted based on updated particle variables at each new time step. Since the ad-

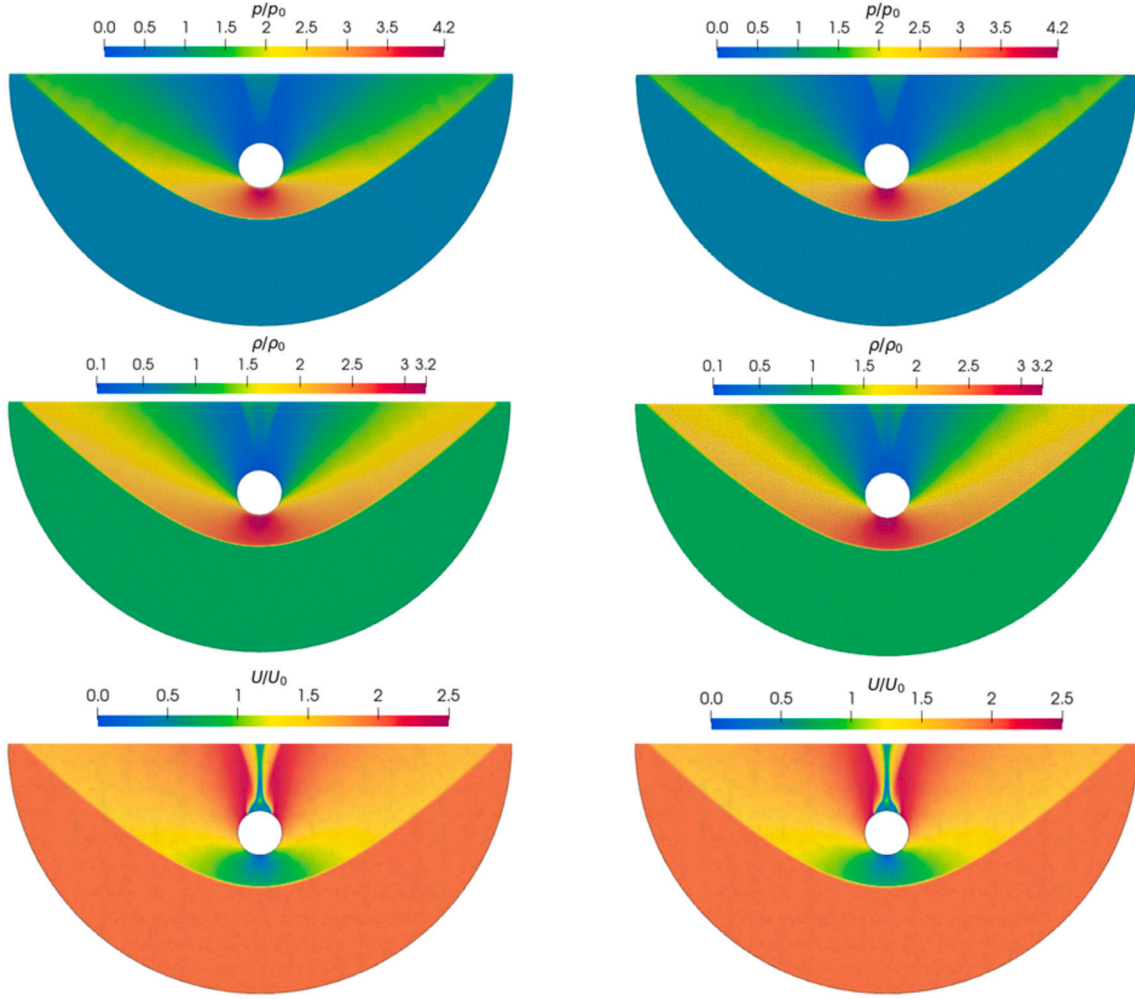


Fig. 10. Supersonic flow past a circular cylinder (Mach 2): The pressure (p/p_0) contour ranging from 0.0 to 4.2 and density (ρ/ρ_0) contour ranging from 0.1 to 3.2 and velocity magnitude (U/U_0) distribution ranging from 0 to 2.5 using the original (left panel) and present (right panel) boundary algorithms at the spatial resolution $dp/D = 1/40$.

ditional storage contains the specific boundary condition and $\mathbf{A}_{ij}^{ESPH}$, the total flux can therefore be calculated from all neighboring particles and the boundary surface, facilitating the update of particle variables at each new time step. Note again, compared with the original algorithm, the boundary conditions are stored and used in the additional storage within the inner configuration in the present algorithm. This approach reduces the need for neighboring particles in the boundaries, thereby decreasing computational effort. While both the original and present algorithms use the Verlet scheme [51,52,26] for time integration, there is a slight difference: for time-dependent boundary conditions, the data in the additional storage, such as velocity in the rigid wall boundary, must be updated at each time step in the present algorithm after all particle variables have been updated.

4. Numerical results

This section examines several numerical examples involving compressible and incompressible flows with complex geometries to validate the stability and accuracy of the proposed boundary algorithm under different boundary conditions in the Eulerian SPH method. In all cases, the Wendland kernel [53] with a smoothing length $h = 1.3dp$ is applied, where dp is the initial particle spacing. For clarity, the original and present boundary algorithms, denoted as “Original” and “Present” for curves among the figures, are applied across all numerical results to

analyze the proposed method straightforwardly and clearly. The reference density ρ_0 , the reference velocity U_0 , and the reference length L_0 are used in simulations.

4.1. Double Mach reflection of a strong shock

In this section, an inviscid compressible flow, i.e., double Mach reflection of a strong shock, is tested using both the FVM and Eulerian SPH to assess the performance of the boundary algorithm for handling various boundary conditions. In this case, the traditional FVM is utilized in SPHInXsys [26] with an unstructured mesh. Following Woodward et al. [54], the computational domain is defined as $(x/L_0, y/L_0) \in [0, 4] \times [0, 1]$, with boundaries including inflow, outflow, shock wave motion, and reflecting boundary conditions, as shown in Fig. 3. Specifically, the bottom boundary is defined as a reflecting wall, initiating at $x/L_0 = 1/6$ where the shock, with a specific heat ratio γ of 1.4, is inclined at a 60-degree angle with respect to the x -axis. It extends vertically to the top of the domain at $y/L_0 = 1$. The initial post-shock input boundary condition is applied from $x/L_0 = 0$ to $x/L_0 = 1/6$. The left-hand boundary is set as the assigned value as the initial post-shock boundary condition, while the right-hand boundary is set as the zero-gradient condition. Specifically, reference parameters are set as follows: $\rho_0 = 1.165 \text{ kg/m}^3$, $U_0 = 294.9 \text{ m/s}$, $L_0 = 1 \text{ m}$, and the corresponding reference pressure $p_0 = 101315 \text{ Pa}$ for compressible flow cases [55,56]. Moreover, a motion boundary condition for the initial Mach 10 shock is

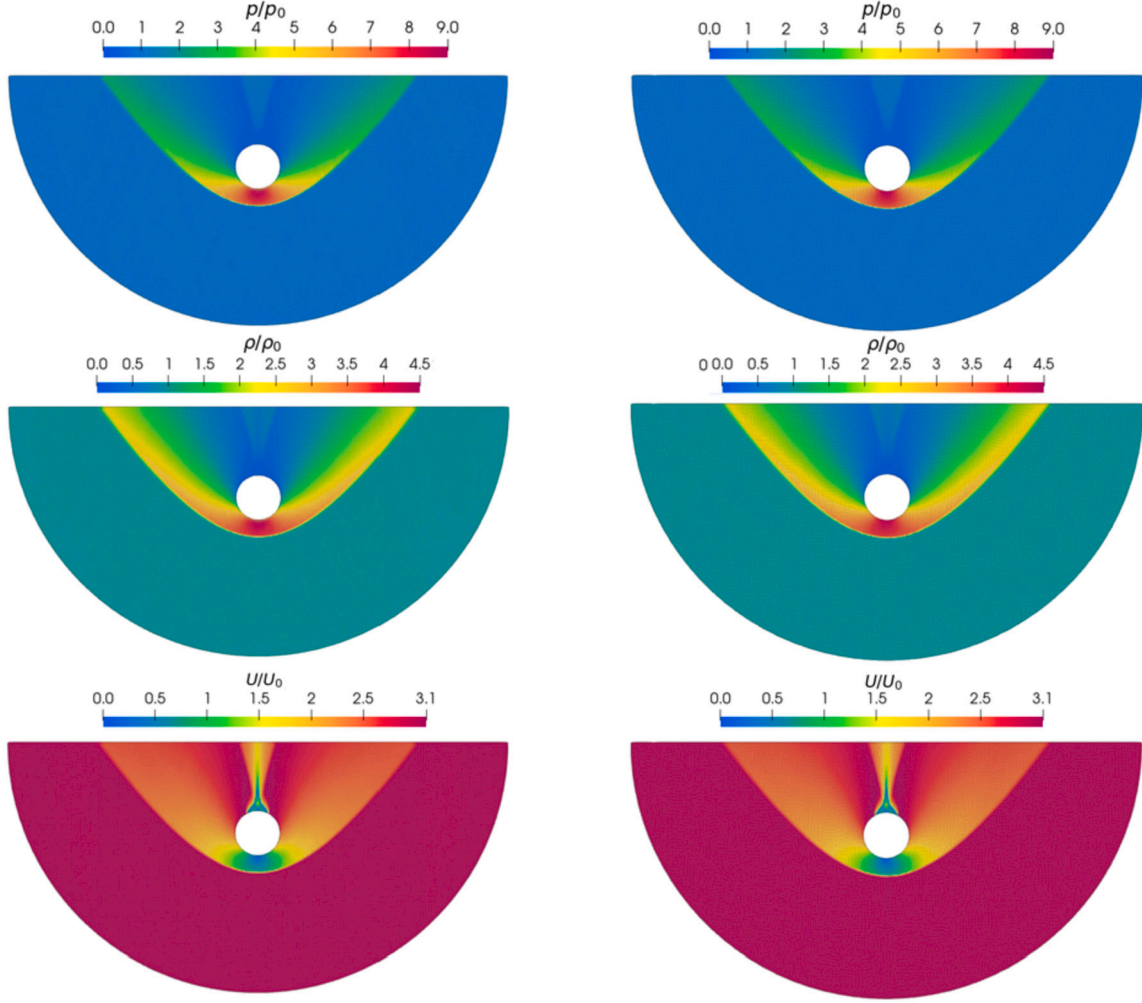


Fig. 11. Supersonic flow past a circular cylinder (Mach 3): The pressure (p/p_0) contour ranging from 0.0 to 9.0 and density (ρ/ρ_0) contour ranging from 0 to 4.5 and velocity magnitude (U/U_0) distribution ranging from 0 to 3.1 using the original (left panel) and present (right panel) boundary algorithms at the spatial resolution $dp/D = 1/40$.

applied to the top boundary. Besides, the initial condition specified as follows:

$$(\rho/\rho_0, u/U_0, v/U_0, p/p_0) = \begin{cases} (1.4, 0, 0, 1) & y/L_0 \leq 1.732(x/L_0 - 0.1667) \\ (8, 7.145, -4.125, 116.8333) & \text{otherwise} \end{cases}, \quad (30)$$

with the time $tU_0/L_0 = 0.2$. In the Eulerian SPH method, the resolution is $dp/L_0 = 1/128$, corresponding to approximately 6.5×10^4 particles, whereas, in FVM, the computational domain is discretized with $N = 6.4 \times 10^4$ cells.

Fig. 4 presents the density contour obtained by the original (also implemented in Refs. [51,26]) and present boundary algorithms with the resolution $dp/L_0 = 1/128$, validating that both boundary algorithms' results are smooth and consistent. Fig. 5 shows the density contour obtained by FVM with a total cell number $N = 64162$, and a zoom-in view for clearer visualization of the unstructured mesh distribution, at the computed time $tU_0/L_0 = 0.2$. By comparing Fig. 5 with Fig. 4, the result can demonstrate that the Eulerian SPH method, coupled with the proposed boundary algorithm, effectively reproduces the result obtained by traditional FVM, characterized by the consistent shock fronts and shear layer. Furthermore, Fig. 6 presents a quantitative comparison of density profiles along $y/L_0 = 0$, $y/L_0 = 0.35$, and $x/L_0 = 1.5$, as indicated by the orange dashed lines in Fig. 5. The results demonstrate that the outcomes using the proposed algorithm are consistent with other results,

validating the accuracy of the proposed boundary algorithm for simulating compressible flows. It is worth noting that the slight differences between the Eulerian SPH and FVM methods may be attributed to the anisotropic properties of the mesh-based FVM and the isotropic characteristics of the particle-based Eulerian SPH, as discussed by Wang et al. [26].

4.2. A Mach 3 wind tunnel with a step

In this section, we examine a Mach 3 inviscid compressible flow around a forward-facing step to validate the effectiveness of the proposed algorithm. As in Refs. [54,57–59], the wind tunnel is one unit in width and three units in length. A step is positioned within this tunnel, located 0.6 units from the tunnel's left-hand extremity, with a height of 0.2 units. The left and right boundaries are set as inflow and supersonic outflow conditions, respectively, and the top and bottom boundaries are set as reflecting boundary conditions, as shown in Fig. 7.

Fig. 8 illustrates the density contours obtained using the original and present boundary algorithms with a spatial resolution of $dp/L_0 = 1/200$ at various time instants. This figure demonstrates the remarkable smoothness of the results obtained by the proposed boundary algorithm, affirming the robustness of the algorithm. For a quantitative analysis, the Mach stem, located around $x/L_0 = 0.8$ displayed with a red dashed line in Fig. 8, is approximately 0.04 units long, and the reflected wave-front strikes the bottom wall, labeled as "s", at about $x/L_0 = 1.715$ at

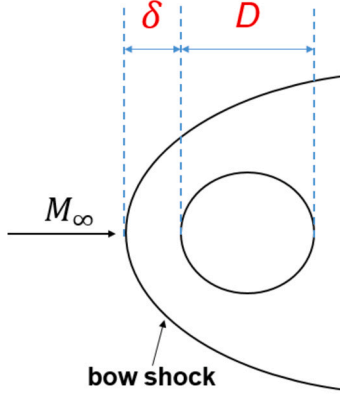


Fig. 12. Supersonic flow past a circular cylinder: The illustration of the shock standoff distance.

Table 1

Supersonic flow past a circular cylinder: The non-dimensional shock standoff distance δ/D obtained from the original and present boundary algorithms, alongside the theoretical solution and other numerical results [62].

	δ/D			
	Theoretical [62]	Sinclair et al. [62]	Original	Present
Mach = 2	0.65	0.68	0.66	0.66
Mach = 3	0.37	0.34	0.35	0.35

$tU_0/L_0 = 2$. These results are in good agreement with those presented by Villodi et al. [60], where the Mach stem is 0.04 units long and the reflected wavefront is located at $x/L_0 = 1.71$, indicating that the proposed boundary algorithm accurately captures the shock waves.

4.3. Supersonic flow past a circular cylinder

In this section, we further explore supersonic compressible flow with complex geometry, namely supersonic flow past a circular cylinder, to evaluate the performance of the proposed algorithm in handling intricate geometries with boundary conditions, including non-reflecting and reflecting boundaries. The reflecting boundary condition is imposed on the cylinder surface, while non-reflecting boundary conditions, including supersonic inflow and outflow, are enforced at the other boundaries [61,62]. The pressure and density inflows are set to $\rho_\infty = \rho_0$ and $p_\infty/p_0 = 1/\gamma$ with $\gamma = 1.4$, respectively. Additionally, the computational domain is defined by a semi-circular geometry with a radius of $5.5D$, where $D = 2L_0$ represents the cylinder diameter. The center of the inner cylinder is located $3.5D$ from the inflow boundary, as illustrated in Fig. 9. In this case, the spatial resolution of $dp/D = 1/40$ is applied to discretize the computational domain.

Figs. 10 and 11 show the pressure, density and velocity contours using the original and present boundary algorithms with Mach = 2 and Mach = 3, respectively. These results demonstrate the stability of the proposed boundary algorithm, which effectively produces smooth results and captures shock properties consistent with the original boundary algorithm and reference results [61,62]. For the quantitative analysis, the non-dimensional shock standoff distance δ/D , displayed in Fig. 12, obtained from the original and present boundary algorithms, as well as the theoretical solution [62] and other numerical results [62], are listed in Table 1, demonstrating the proposed boundary algorithm is accurate. Here, the code for this case is available at https://github.com/Xiangyu-Hu/SPHinXsys/tree/master/tests/2d_examples/test_2d_eulerian_supersonic_flow_new_BC.

4.4. Poiseuille flow

In this section, the 2D Poiseuille flow, a fully developed incompressible flow in a channel, is used as a benchmark for the quantitative assessment of the accuracy of the proposed boundary algorithm. For the incompressible flow cases, reference parameters are set as follows: $\rho_0 = 10^3 \text{ kg/m}^3$, $U_0 = 1 \text{ m/s}$, and $L_0 = 1 \text{ m}$. Following Liu et al. [63], the Reynolds number $Re = U_{max}H/\nu$, with ν denoting kinematic viscosity, is defined with the channel height $H = L_0$, the maximum velocity $U_{max} = U_0$, and the pressure gradient is set as $\partial p/\partial x = -8\rho\nu U_{max}/H^2$. Additionally, the analytical solution is defined as

$$U_{analytical} = U_{max}(1 - \frac{y^2}{H^2}), \quad (31)$$

where y represents the distance from the horizontal centerline at $0.5H$. A periodic boundary is employed along the x -axis, while a non-slip boundary is applied to the remaining boundaries with the original and present boundary algorithms.

Fig. 13 shows the velocity distribution using the original and present boundary algorithms, demonstrating the smoothness of the results. Besides, Fig. 14 presents the velocity profile along the x -axis direction as a function of the y position, showing good agreement with the analytical solution for both algorithms. Furthermore, Fig. 15 presents the L_2 error [64], defined by

$$L_2 = \frac{\left(\sum_i |U_{analytical} - U_{numerical}|^2\right)^{1/2}}{\left(\sum_i |U_{analytical}|^2\right)^{1/2}}, \quad (32)$$

with $U_{numerical}$ representing the SPH approximated result, and the convergence analysis. The theoretical error convergence rate of the SPH approximation with a smooth kernel is second-order. However, in practice, the observed convergence rates often decrease to 1.5 or lower [65]. In Fig. 15, the original boundary algorithm maintains a convergence rate of approximately 1.5. In contrast, the present boundary algorithm achieves a second-order convergence rate at high resolutions, indicating an enhancement in the convergence rate. Furthermore, the present boundary algorithm demonstrates numerical accuracy comparable to or surpassing the original algorithm's.

4.5. Lid-driven cavity flows with different shapes

In this section, we investigate incompressible lid-driven cavity flows characterized by different shapes to assess the stability and accuracy of the proposed boundary algorithm in complex geometries. We begin by examining a simple square cavity, comparing results obtained using both the original boundary algorithm (as implemented in Refs. [51,26]) and the present approach against the reference results from Ghia et al. [66] to verify numerical accuracy. Additionally, a semi-circular cavity is tested to investigate the algorithm's capability in identifying and implementing boundaries within complex geometries by comparing it with the reference result from Glowinski et al. [67]. For the boundary conditions, the upper moving wall is assigned a prescribed velocity of $U_{wall} = U_0$, while non-slip conditions are applied to the remaining boundaries.

4.5.1. Lid-driven square cavity flow

For the square cavity, the computational domain is a square of length $L = L_0$ with the Reynolds number $Re = 400$. Fig. 16 displays the velocity magnitude distribution obtained by the original and present boundary algorithms at a resolution of $dp/L = 1/129$. The results reveal that the velocity fields obtained with both algorithms exhibit good smoothness. Besides, Fig. 17 presents the horizontal velocity component u/U_0 along $x = 0.5L$ and the vertical velocity component v/U_0 along $y = 0.5L$, using the original and present boundary algorithms with different resolutions, and the comparisons with the reference by Ghia et al.

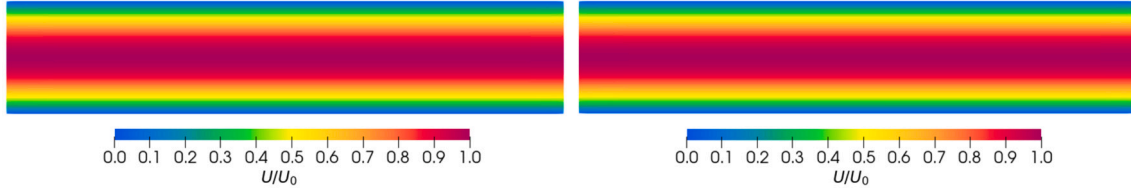


Fig. 13. Poiseuille flow: The velocity magnitude (U/U_0) distribution using the original (left panel) and present (right panel) boundary algorithms ranging from 0 to 1 with the spatial resolution $dp/H = 1/80$.

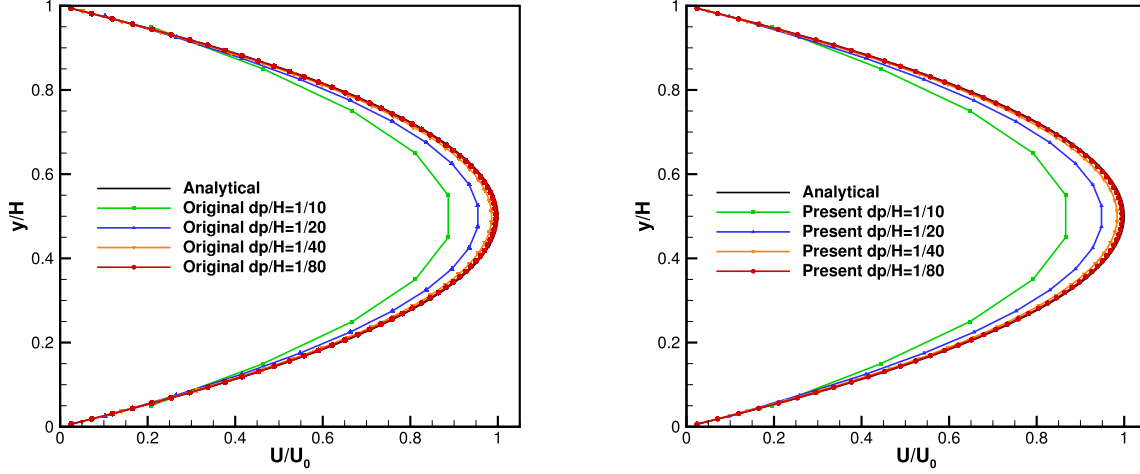


Fig. 14. Poiseuille flow: The velocity magnitude (U/U_0) profile obtained by the original (left) and present (right) boundary algorithms with the spatial resolutions as $dp/H = 1/10, 1/20, 1/40$ and $1/80$ as well as the analytical solution.

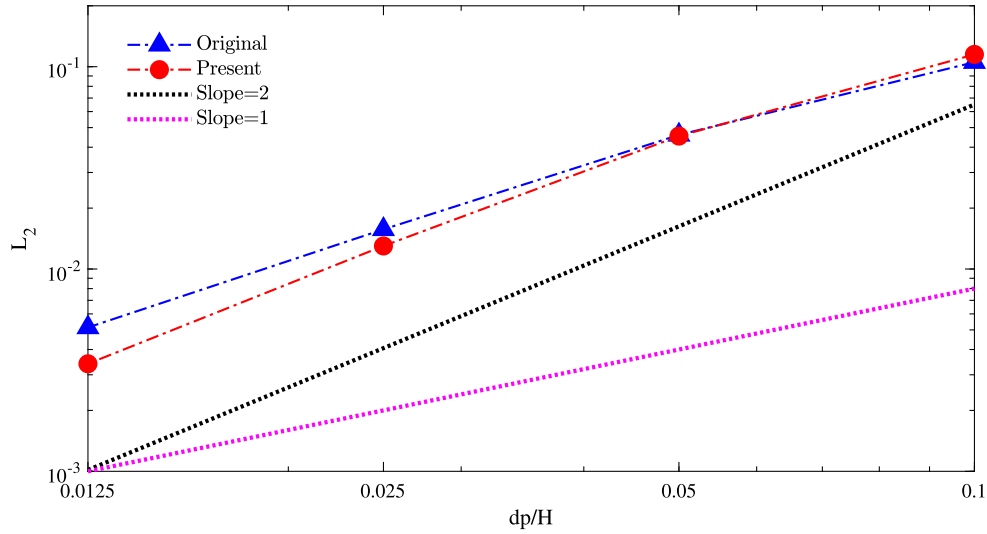


Fig. 15. Poiseuille flow: L_2 error and convergence analysis using the original and present boundary algorithms.

[66]. The outcomes demonstrate that the present algorithm achieves numerical accuracy comparable to the original algorithm and the reference result and that the results converge rapidly with increasing resolutions.

4.5.2. Lid-driven semi-circular cavity flow

Following Glowinski et al. [67], the circle's diameter is $L = L_0$, and the Reynolds number is set as $Re = 1000$ for the semi-circular cavity flow case. Fig. 18 shows the velocity contours using the original and present boundary algorithms with the spatial resolution $dp/L = 1/129$, both demonstrating the velocity fields' smoothness. Additionally, Fig. 19 presents the horizontal dimensionless velocity component u/U_0 along $x = 0.5L$ and the vertical dimensionless velocity component v/U_0 along $y = -0.25L$ using the original boundary algorithm with the resolution

of $dp/L = 1/129$ and the present boundary algorithm with the resolutions of $dp/L = 1/129$ and $1/257$, as well as the reference results [67]. This comparison confirms that both boundary algorithms achieve comparable accuracy. Furthermore, due to the introduction of the Riemann problem, the results using the resolution of $dp/L = 1/257$ are consistent with the reference results.

4.6. Flow around a circular cylinder

In this section, we investigate the impact of the proposed boundary algorithm in a viscous incompressible flow, i.e., the flow around a circular cylinder. This problem is characterized by complex geometry and incorporates the boundary conditions, including non-slip wall and

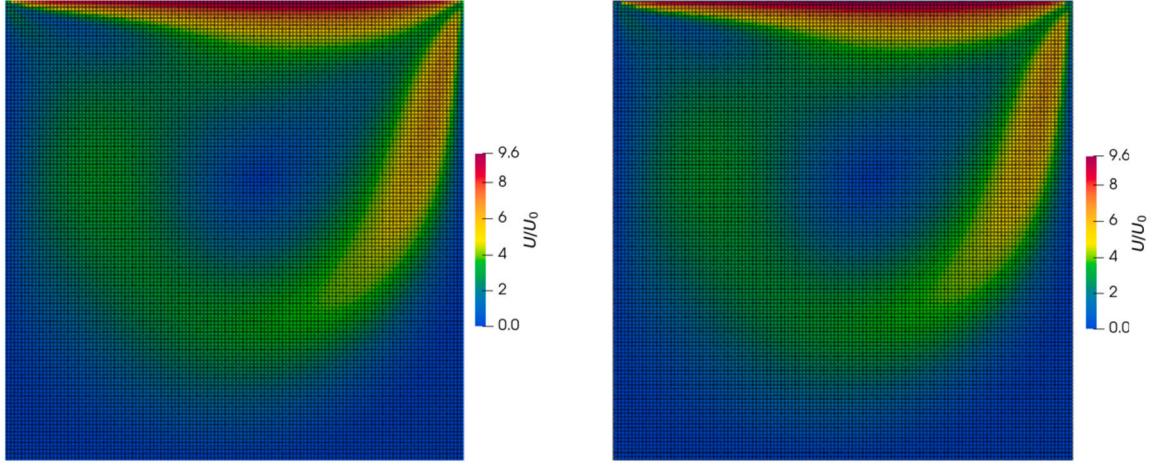


Fig. 16. Lid-driven cavity flow ($Re = 400$): The velocity magnitude (U/U_0) contour using the original (left panel) and present (right panel) boundary algorithms ranging from 0 to 0.96 with the resolution as $dp/L = 1/129$.

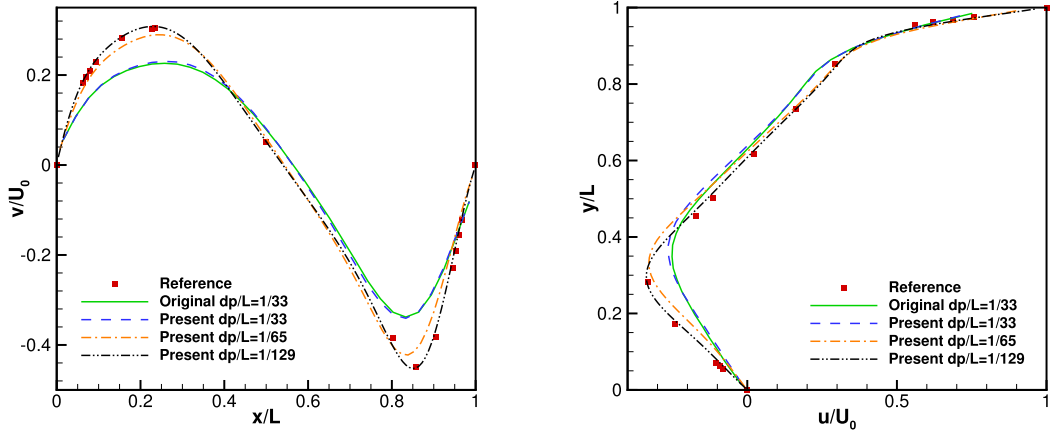


Fig. 17. Lid-driven cavity flow ($Re = 400$): The horizontal velocity component u/U_0 along $x = 0.5L$ (left panel) and the vertical velocity component v/U_0 along $y = 0.5L$ (right panel) using the original boundary algorithm with the resolution of $dp/L = 1/33$ and the present boundary algorithm with the resolutions of $dp/L = 1/33, 1/65$ and $1/129$, and the comparisons with the reference by Ghia et al. [66].

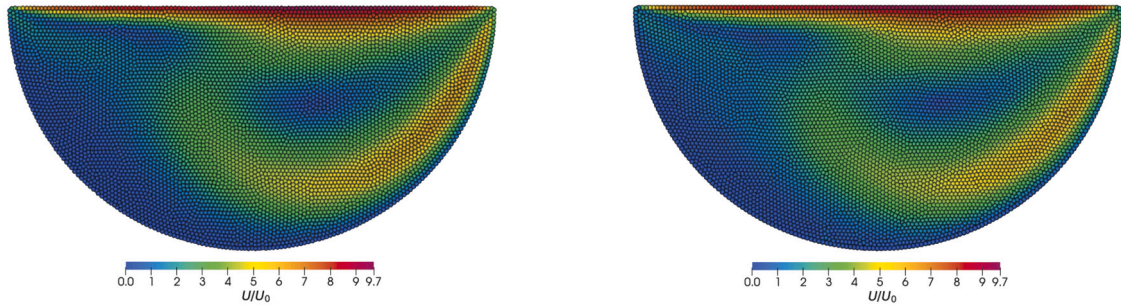


Fig. 18. Lid-driven semi-circular cavity flow ($Re = 1000$): The velocity magnitude (U/U_0) contour using the original (left panel) and present (right panel) boundary algorithms ranging from 0 to 0.97 with the spatial resolution $dp/L = 1/129$.

far-field boundary conditions. As illustrated in Fig. 20, the circular cylinder is assigned a no-slip wall boundary condition, while the remaining boundaries are set as far-field boundary conditions. A computational domain size of $[25D, 15D]$ is employed, where the center of the cylinder is positioned at $(7.5D, 7.5D)$ and a cylinder diameter of $D = 2L_0$. To quantitatively analyze the accuracy of the results, we employ the drag and lift coefficients, which are defined as:

$$C_D = \frac{2F_D}{\rho_\infty u_\infty^2 A}, C_L = \frac{2F_L}{\rho_\infty u_\infty^2 A}, \quad (33)$$

where F_D and F_L represent the drag and lift forces on the cylinder, respectively, and A , equal to D , is a reference area. Besides, a constant far-field velocity is $u_\infty = U_0$ and density $\rho_\infty = \rho_0$, and the Strouhal number $St = fD/u_\infty$, where f is the frequency in the unsteady cases. In the case, the Reynolds numbers $Re = \rho_\infty u_\infty D/\mu$ are set as 20, 100 and 200, and the computational time is set as $tU_0/L_0 = 300$.

Fig. 21 portrays the velocity distributions using the original (also implemented in Wang et al. [51]) and present boundary algorithms for Reynolds numbers of $Re = 20, 100$ and 200 at the final time $tU_0/L_0 = 300$. The results demonstrate smooth and consistent outcomes for both

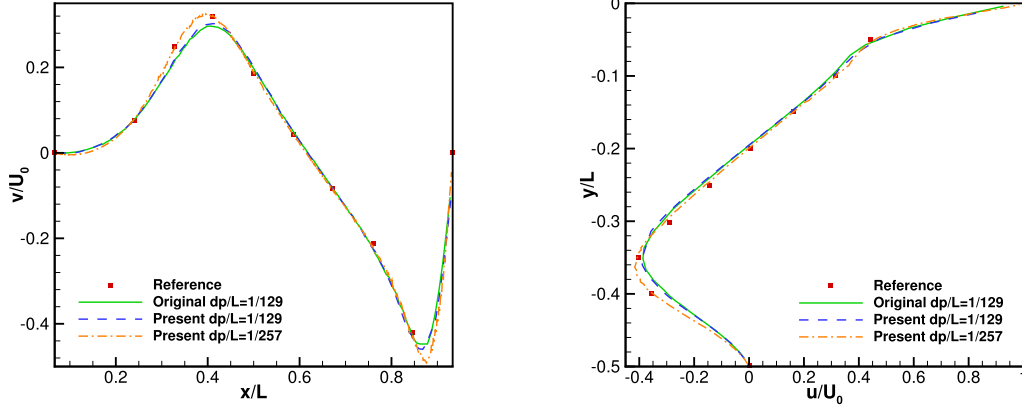


Fig. 19. Lid-driven semi-circular cavity flow ($Re = 1000$): The horizontal velocity component u/U_0 along $x = 0.5L$ (left panel) and the vertical velocity component v/U_0 along $y = -0.25L$ (right panel) using the original with $dp/L = 1/129$ and present boundary algorithms with $dp/L = 1/129$ and $1/257$, and the comparison with the reference obtained by Glowinski et al. [67].

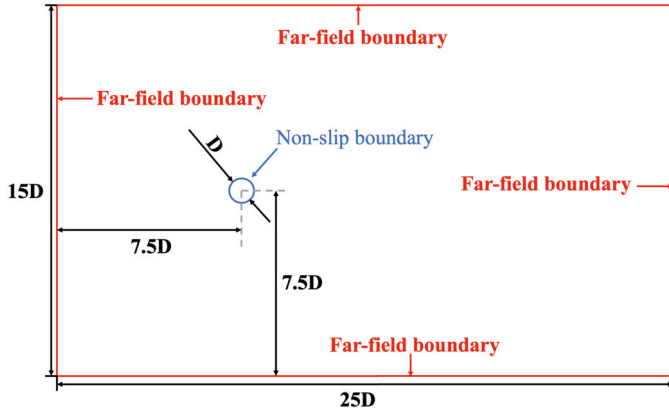


Fig. 20. Flow around a circular cylinder: The geometry and boundary conditions.

Table 2

Flow around a cylinder: Drag and lift coefficients from different experimental and numerical results with $Re = 20$ and the result using the present boundary algorithm with the resolution $dp/D = 1/60$.

	C_D
Tritton [68]	2.09
Taira and Colonius [69]	2.06
Tafuni et al. [70]	2.29
Negi et al. [71]	2.32
Present	2.32

Table 3

Flow around a cylinder: Drag and lift coefficients from different experimental and numerical results with $Re = 100$ and the result using the present boundary algorithm with the resolution $dp/D = 1/60$.

	C_D	C_L	S_i
White [72]	1.46	-	-
Chiu et al. [73]	1.35 ± 0.012	± 0.303	0.166
Le et al. [74]	1.37 ± 0.009	± 0.323	0.160
Brehm et al. [75]	1.32 ± 0.010	± 0.320	0.165
Liu et al. [76]	1.35 ± 0.012	± 0.339	0.165
Zhang et al. [49]	1.61 ± 0.005	± 0.448	0.171
Present	1.46 ± 0.009	± 0.349	0.176

Table 4

Flow around a cylinder: Drag and lift coefficients from different experimental and numerical results with $Re = 200$ and the result using the present boundary algorithm with the resolution $dp/D = 1/60$.

	C_D	C_L	S_i
Taira et al. [69]	1.35 ± 0.048	± 0.68	0.196
Tafuni et al. [70]	1.46	± 0.693	0.206
Negi et al. [71]	1.524 ± 0.05	± 0.722	0.210
Jin and Braza [77]	1.532 ± 0.05	± 0.744	0.210
Zhang et al. [49]	1.63 ± 0.05	± 0.84	0.212
Present	1.48 ± 0.055	± 0.806	0.208

algorithms. Besides, the converged results are listed in Tables 2, 3 and 4, which contains a series of numerical and experimental results for different Reynolds numbers $Re = 20$, 100 and 200, respectively. The observations show that the converged results agree with the references, validating that the proposed algorithm is reliable and accurate. As GPU acceleration has not been implemented in the present version of SPHinXsys, the comparison of computational efficiency is limited to CPU performance. This study executed the computations on a desktop computer with an Intel Core i7-10700 processor featuring eight cores and a clock speed of 2.90 GHz. The total CPU wall-clock time for the original boundary algorithm [26], with the spatial resolution of $dp/D = 1/20$ and the Reynolds number of $Re = 100$, amounts to 4141.94 seconds. However, the wall-clock time required by the present boundary algorithm is 3600.63 seconds, demonstrating its greater computational efficiency.

5. Summary and conclusion

In this paper, we implement a particle relaxation technique to generate a body-fitted particle distribution in which particles not in the boundaries satisfy zero-order consistency, and also introduce low-dissipation limiters and the kernel correction to enhance numerical accuracy. By analyzing the boundary algorithm in FVM, we propose an algorithmically equivalent approach within the Eulerian SPH framework, ensuring that particles in the boundaries strictly achieve zero-order consistency. Moreover, several numerical cases, including compressible and incompressible flows, have been tested, validating the comparable stability and accuracy as well as an enhanced convergence rate of the present boundary algorithm compared with the original algorithm. Another supplementary advantage is the improvement in computational efficiency.

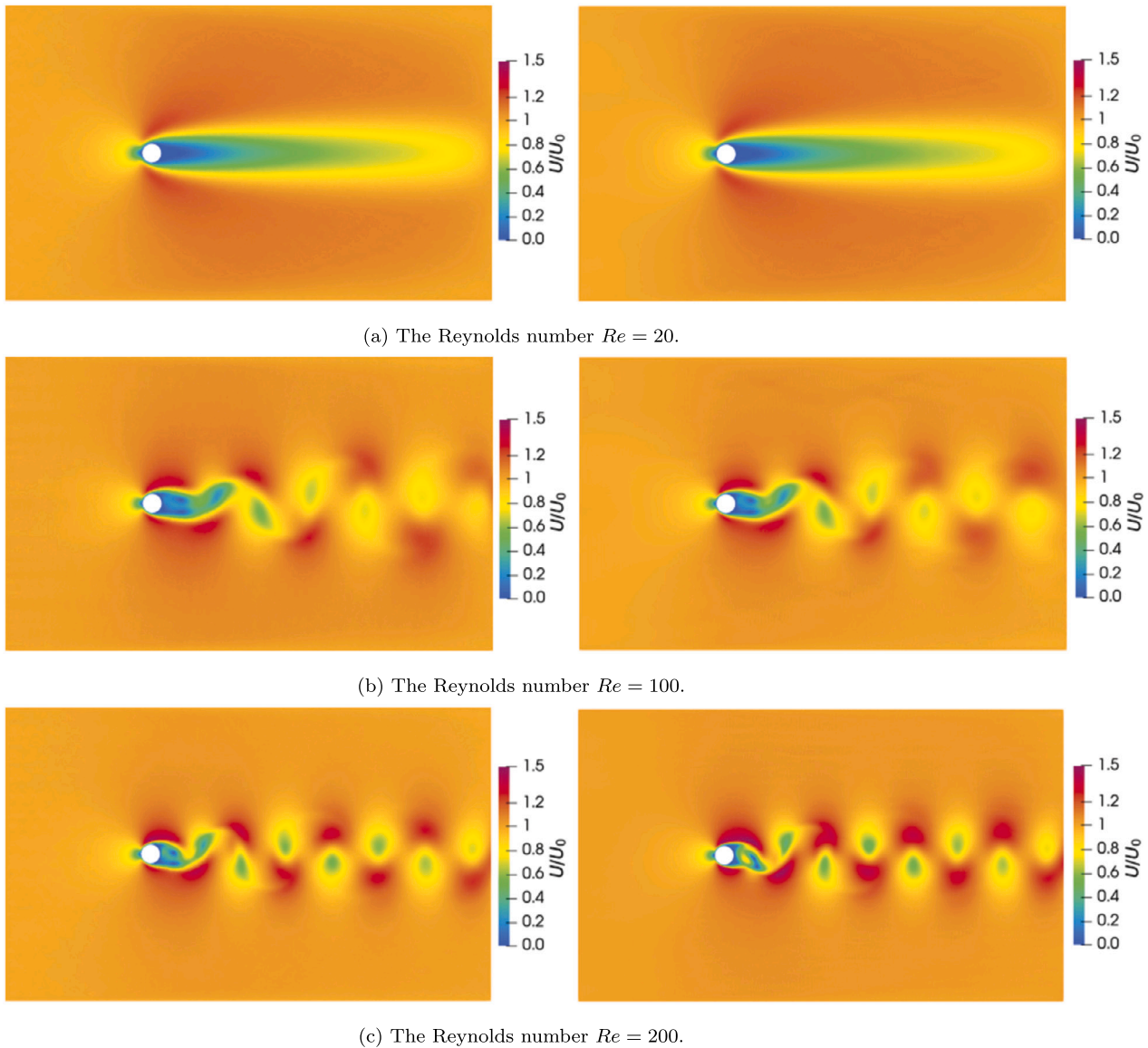


Fig. 21. Flow around a circular cylinder: The velocity magnitude (U/U_0) distribution using the original (left panel) and present (right panel) boundary algorithms ranging from 0 to 1.5 with the different Reynolds numbers at the final time $tU_0/L_0 = 300$.

CRediT authorship contribution statement

Zhentong Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis. **Oskar J. Haidn:** Writing – review & editing, Supervision, Investigation. **Xiangyu Hu:** Writing – review & editing, Supervision, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code is open source.

References

- [1] F. Afshari, H.G. Zavaragh, B. Sahin, R.C. Grifoni, F. Corvaro, B. Marchetti, F. Polonara, On numerical methods; optimization of CFD solution to evaluate fluid flow around a sample object at low Re numbers, *Math. Comput. Simul.* 152 (2018) 51–68.
- [2] J. O'connor, J.M. Domínguez, B.D. Rogers, S.J. Lind, P.K. Stansby, Eulerian incompressible smoothed particle hydrodynamics on multiple GPUs, *Comput. Phys. Commun.* 273 (2022) 108263.
- [3] K. Mahady, S. Afkhami, L. Kondic, A volume of fluid method for simulating fluid/fluid interfaces in contact with solid boundaries, *J. Comput. Phys.* 294 (2015) 243–257.
- [4] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (1) (2001) 132–150.
- [5] F. Roman, E. Napoli, B. Milici, V. Armenio, An improved immersed boundary method for curvilinear grids, *Comput. Fluids* 38 (8) (2009) 1510–1527.
- [6] C. Erath, A posteriori error estimates and adaptive mesh refinement for the coupling of the finite volume method and the boundary element method, *SIAM J. Numer. Anal.* 51 (3) (2013) 1777–1804.
- [7] S. Cheng, M. Zhang, Z. Chen, B. Wu, Numerical study of simultaneous growth of multiple hydraulic fractures from a horizontal wellbore combining dual boundary element method and finite volume method, *Eng. Anal. Bound. Elem.* 139 (2022) 278–292.

- [8] Z.-Y. Li, Z.-J. Chen, X.-H. Wu, W.-Q. Tao, Coupled MLPG-FVM simulation of steady state heat conduction in irregular geometry, *Eng. Anal. Bound. Elem.* 100 (2019) 265–275.
- [9] L. Chiron, S. Marrone, A. Di Mascio, D. Le Touzé, Coupled sph-fv method with net vorticity and mass transfer, *J. Comput. Phys.* 364 (2018) 111–136.
- [10] E. Napoli, M. De Marchis, C. Gianguzzi, B. Milici, A. Monteleone, A coupled finite volume-smoothed particle hydrodynamics method for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 310 (2016) 674–693.
- [11] A. Di Mascio, S. Marrone, A. Colagrossi, L. Chiron, D. Le Touzé, SPH-FV coupling algorithm for solving multi-scale three-dimensional free-surface flows, *Appl. Ocean Res.* 115 (2021) 102846.
- [12] Y. Xu, G. Yang, Y. Zhu, D. Hu, A coupled SPH-FVM method for simulating incompressible interfacial flows with large density difference, *Eng. Anal. Bound. Elem.* 128 (2021) 227–243.
- [13] F. Chen, H. Qiang, W. Gao, Coupling of smoothed particle hydrodynamics and finite volume method for two-dimensional spouted beds, *Comput. Chem. Eng.* 77 (2015) 135–146.
- [14] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (3) (1977) 375–389.
- [15] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (Dec. 1977) 1013–1024.
- [16] J.J. Monaghan, Simulating free surface flows with SPH, *J. Comput. Phys.* 110 (2) (1994) 399–406.
- [17] Y. Shimizu, A. Khayyer, H. Gotoh, K. Nagashima, An enhanced multiphase ISPH-based method for accurate modeling of oil spill, *Coast. Eng. J.* 62 (4) (2020) 625–646.
- [18] A. Khayyer, H. Gotoh, Y. Shimizu, T. Gotoh, An improved Riemann SPH-Hamiltonian SPH coupled solver for hydroelastic fluid-structure interactions, *Eng. Anal. Bound. Elem.* 158 (2024) 332–355.
- [19] L.D. Libersky, A.G. Petschek, Smooth particle hydrodynamics with strength of materials, in: *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method: Proceedings of the Next Free-Lagrange Conference Held at Jackson Lake Lodge, Moran, WY, USA 3–7 June 1990*, Springer, 2005, pp. 248–257.
- [20] C. Zhang, Y.-j. Zhu, D. Wu, N.A. Adams, X. Hu, Smoothed particle hydrodynamics: methodology development and recent achievement, *J. Hydrodyn.* 34 (5) (2022) 767–805.
- [21] H. Gotoh, A. Khayyer, Y. Shimizu, Entirely Lagrangian meshfree computational methods for hydroelastic fluid-structure interactions in ocean engineering—reliability, adaptivity and generality, *Appl. Ocean Res.* 115 (2021) 102822.
- [22] S. Wang, J. Hu, C. Huang, Y. Yu, Graphics processing unit-accelerated smoothed particle hydrodynamics—finite difference method and the application for the flow around a cylinder with forced motions, *Phys. Fluids* 33 (12) (2021).
- [23] M.-K. Li, A. Zhang, F.-R. Ming, Y.-X. Peng, A coupled smoothed particle hydrodynamics-finite volume method for three-dimensional modeling of bubble dynamics, *Phys. Fluids* 35 (5) (2023).
- [24] S. Marrone, A. Di Mascio, D. Le Touzé, Coupling of smoothed particle hydrodynamics with finite volume method for free-surface flows, *J. Comput. Phys.* 310 (2016) 161–180.
- [25] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B.D. Rogers, A. Souto-Iglesias, Grand challenges for smoothed particle hydrodynamics numerical schemes, *Comput. Part. Mech.* 8 (2021) 575–588.
- [26] Z. Wang, C. Zhang, O.J. Haidn, N.A. Adams, X. Hu, Extended Eulerian SPH and its realization of FVM, *arXiv preprint, arXiv:2309.01596*, 2023.
- [27] C. Zhang, M. Rezavand, Y. Zhu, Y. Yu, D. Wu, W. Zhang, J. Wang, X. Hu, SPHinXsys: an open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics, *Comput. Phys. Commun.* 267 (2021) 108066.
- [28] C. Zhang, M. Rezavand, Y. Zhu, Y. Yu, D. Wu, W. Zhang, S. Zhang, J. Wang, X. Hu, SPHinXsys: an open-source meshless, multi-resolution and multi-physics library, *Softw. Impacts* 6 (2020) 100033.
- [29] C. Zhang, X. Hu, N.A. Adams, A weakly compressible SPH method based on a low-dissipation Riemann solver, *J. Comput. Phys.* 335 (2017) 605–620.
- [30] J.P. Vila, On particle weighted methods and smooth particle hydrodynamics, *Math. Models Methods Appl. Sci.* 9 (02) (1999) 161–209.
- [31] B. Ben Moussa, On the convergence of SPH method for scalar conservation laws with boundary conditions, *Methods Appl. Anal.* 13 (1) (2006) 29–62.
- [32] P. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, *Comput. Methods Appl. Mech. Eng.* 139 (1–4) (1996) 375–408.
- [33] V. Zago, L.J. Schulze, G. Bilotta, N. Almashan, R. Dalrymple, Overcoming excessive numerical dissipation in SPH modeling of water waves, *Coast. Eng.* 170 (2021) 104018.
- [34] Y. Zhu, C. Zhang, Y. Yu, X. Hu, A CAD-compatible body-fitted particle generator for arbitrarily complex geometry and its application to wave-structure interaction, *J. Hydrodyn.* 33 (2) (2021) 195–206.
- [35] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL- Riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [36] E.F. Toro, The hllc Riemann solver, *Shock Waves* 29 (8) (2019) 1065–1082.
- [37] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*, Springer Science & Business Media, 2013.
- [38] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* 32 (1) (1979) 101–136.
- [39] C. Zhang, G. Xiang, B. Wang, X. Hu, N.A. Adams, A weakly compressible SPH method with WENO reconstruction, *J. Comput. Phys.* 392 (2019) 1–18.
- [40] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. Comput. Phys.* 54 (1) (1984) 174–201.
- [41] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *Int. J. Numer. Methods Eng.* 66 (13) (2006) 2064–2085.
- [42] J. Domínguez, A. Crespo, A. Barreiro, M. Gómez-Gesteira, A. Mayrhofer, Development of a new pre-processing tool for SPH models with complex geometries, in: *6th International SPHERIC Workshop*, 2011, pp. 117–124.
- [43] A. Colagrossi, B. Bouscasse, M. Antuono, S. Marrone, Particle packing algorithm for SPH schemes, *Comput. Phys. Commun.* 183 (8) (2012) 1641–1653.
- [44] L.V. Vela, R. Sanchez, J. Geiger, ALARIC: an algorithm for constructing arbitrarily complex initial density distributions with low particle noise for SPH/SPMHD applications, *Comput. Phys. Commun.* 224 (2018) 186–197.
- [45] L. Fu, Z. Ji, An optimal particle setup method with centroidal Voronoi particle dynamics, *Comput. Phys. Commun.* 234 (2019) 72–92.
- [46] A. Khayyer, C.H. Lee, Discussion on “Comparative study on volume conservation among various SPH models for flows of different levels of violence, Coastal Engineering, Volume 191, August 2024, 104521” by Wang et al., *Coast. Eng.* (2024) 104565.
- [47] M. Neuhauser, Development of a coupled SPH-ALE/Finite Volume method for the simulation of transient flows in hydraulic machines, Ph.D. thesis, Ecully, Ecole centrale de Lyon, 2014.
- [48] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method, *J. Comput. Phys.* 227 (18) (2008) 8417–8436.
- [49] S. Zhang, W. Zhang, C. Zhang, X. Hu, A Lagrangian free-stream boundary condition for weakly compressible smoothed particle hydrodynamics, *J. Comput. Phys.* (2023) 112303.
- [50] S. Adami, X.Y. Hu, N.A. Adams, A generalized wall boundary condition for smoothed particle hydrodynamics, *J. Comput. Phys.* 231 (21) (2012) 7057–7075.
- [51] Z. Wang, C. Zhang, O.J. Haidn, X. Hu, An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows, *J. Hydrodyn.* (2023) 1–12.
- [52] C. Zhang, M. Rezavand, X. Hu, A multi-resolution SPH method for fluid-structure interactions, *J. Comput. Phys.* 429 (2021) 110028.
- [53] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.* 4 (1995) 389–396.
- [54] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.
- [55] Q. Li, Y. He, Y. Wang, W. Tao, Coupled double-distribution-function lattice Boltzmann method for the compressible Navier-Stokes equations, *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.* 76 (5) (2007) 056705.
- [56] M. Hryniewicki, J. Gottlieb, C. Groth, Transition boundary between regular and Mach reflections for a moving shock interacting with a wedge in inviscid and polytropic air, *Shock Waves* 27 (2017) 523–550.
- [57] C.-Y. Wen, Y. Jiang, L. Shi, Space-Time Conservation Element and Solution Element Method: Advances and Applications in Engineering Sciences, Springer Nature, 2023.
- [58] J. Zhu, J. Qiu, A new type of finite volume WENO schemes for hyperbolic conservation laws, *J. Sci. Comput.* 73 (2017) 1338–1359.
- [59] D. Wang, X. Deng, G. Wang, Y. Dong, Developing a hybrid flux function suitable for hypersonic flow simulation with high-order methods, *Int. J. Numer. Methods Fluids* 81 (5) (2016) 309–327.
- [60] N. Villodi, P. Ramachandran, Robust solid boundary treatment for compressible smoothed particle hydrodynamics, *Phys. Fluids* 36 (8) (2024).
- [61] Y. Zhu, X. Hu, Free-stream preserving linear-upwind and WENO schemes on curvilinear grids, *J. Comput. Phys.* 399 (2019) 108907.
- [62] J. Sinclair, X. Cui, A theoretical approximation of the shock standoff distance for supersonic flows around a circular cylinder, *Phys. Fluids* 29 (2) (2017).
- [63] C.-H. Liu, K.-H. Lin, H.-C. Mai, C.-A. Lin, Thermal boundary conditions for thermal lattice Boltzmann simulations, *Comput. Math. Appl.* 59 (7) (2010) 2178–2193.
- [64] Y. Ren, P. Lin, C. Zhang, X. Hu, An efficient correction method in Riemann SPH for the simulation of general free surface flows, *Comput. Methods Appl. Mech. Eng.* 417 (2023) 116460.
- [65] A.M. Nasar, G. Fourtakas, S.J. Lind, B.D. Rogers, P. Stansby, J. King, High-order velocity and pressure wall boundary conditions in Eulerian incompressible SPH, *J. Comput. Phys.* 434 (2021) 109793.
- [66] U. Ghia, K.N. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.* 48 (3) (1982) 387–411.
- [67] R. Glowinski, G. Guidoboni, T.-W. Pan, Wall-driven incompressible viscous flow in a two-dimensional semi-circular cavity, *J. Comput. Phys.* 216 (1) (2006) 76–91.
- [68] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, *J. Fluid Mech.* 6 (4) (1959) 547–567.
- [69] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 225 (2) (2007) 2118–2137.
- [70] A. Tafuni, J. Domínguez, R. Vacondio, A. Crespo, A versatile algorithm for the treatment of open boundary conditions in smoothed particle hydrodynamics GPU models, *Comput. Methods Appl. Mech. Eng.* 342 (2018) 604–624.
- [71] P. Negi, P. Ramachandran, A. Haftu, An improved non-reflecting outlet boundary condition for weakly-compressible SPH, *Comput. Methods Appl. Mech. Eng.* 367 (2020) 113119.

- [72] F.M. White, J. Majdalani, *Viscous Fluid Flow*, vol. 3, McGraw-Hill, New York, 2006.
- [73] P.-H. Chiu, R.-K. Lin, T.W. Sheu, A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries, *J. Comput. Phys.* 229 (12) (2010) 4476–4500.
- [74] D.-V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.* 220 (1) (2006) 109–138.
- [75] C. Brehm, C. Hader, H.F. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 295 (2015) 475–504.
- [76] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1) (1998) 35–57.
- [77] G. Jin, M. Braza, A nonreflecting outlet boundary condition for incompressible unsteady Navier–Stokes calculations, *J. Comput. Phys.* 107 (2) (1993) 239–253.

A.3 Paper III

Zhentong Wang, Chi Zhang, Oskar J. Haidn, Xiangyu Hu

An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows

In *Journal of Hydrodynamics*, Volume 35, 2023, pp. 210-221, DOI: <https://doi.org/10.1007/s42241-023-0025-z>.

Copyright © 2023 Springer. Reprinted with permission.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.

An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows

Zhentong Wang¹, Chi Zhang^{1,2*}, Oskar J. Haidn¹, Xiangyu Hu¹

1. TUM School of Engineering and Design, Technical University of Munich, Munich, Germany

2. Huawei Technologies Munich Research Center, Munich, Germany

(Received April 20, 2023, Revised May 5, 2023, Accepted May 6, 2023, Published online May 18, 2023)

©China Ship Scientific Research Center 2023

Abstract: While Eulerian smoothed particle hydrodynamics (SPH) method has received increasing attention in scientific and industrial communities owing to its high spatial accuracy, it exhibits excessive numerical dissipation due to the fact that the flux is derived in particle pair pattern. In this paper, we adopt a one-dimensional weighted essentially non-oscillatory (WENO) reconstruction to reduce the numerical dissipation and improve the overall accuracy particularly in capturing the contact discontinuity. The underlying principle is to construct a 4-point stencil along the interacting line of each particle pair and then the WENO scheme is applied to reconstruct the initial states of the Riemann problem which determines the flow flux. A set of benchmark tests for both compressible and incompressible flows are studied to investigate the accuracy, robustness and versatility of the proposed Eulerian SPH method with the WENO reconstruction (ESPH-WENO).

Key words: Eulerian smoothed particle hydrodynamics (SPH), compressible flows, incompressible flows, Riemann problem

0. Introduction

Smooth particle hydrodynamics (SPH) is a pure meshless method and was originally proposed by Lucy^[1], Gingold and Monaghan^[2] for astrophysical applications. Since its inception, the SPH method has also been effectively applied in fluid dynamics^[3], structural mechanics^[4], fluid-structure interaction (FSI)^[5-7] and more details are referred to recent comprehensive review papers^[8-9].

Depending on whether the particle position is fixed, the SPH method is characterized by Eulerian or Lagrangian framework. The Lagrangian framework has natural advantages for handling large deformations such as violent free-surface evolution, multiphase interface and FSI. Although the Lagrangian SPH method has the flexibility of topological variation, it also leads to the problem of poor quality of particle distribution. On the contrary, the Eulerian framework can effectively alleviate this issue due to fixed particle position and obtain high spatial accuracy^[10-12], meanwhile greatly reduce the computational effort^[13] as neighboring particle search is avoided.

For both Eulerian and Lagrangian SPH formula-

tion, numerical dissipation is essential for stabilized simulation. One common approach is to introduce artificial viscosity in the momentum^[14-15], while this approach leads to excessive dissipation which affects the physical flow characteristic and needs tuning parameters. Another common approach is to impose the Riemann problem along the particle interacting line and the numerical dissipation is intrinsically introduced. To achieve higher accuracy, a dissipation limiter is usually applied to reduce the numerical dissipation^[16]. Furthermore, numerical schemes to achieve first-order consistency are also developed to improve the numerical accuracy. A kernel correction technique is introduced to ensure the accurate evaluation of the gradient approximation and achieve first-order consistency^[17-18]. However, both the low-dissipation Riemann solvers and the kernel correction technique provide limited improvement in numerical accuracy for the simulation of both incompressible and compressible flows, and even more limited improvement for the contact discontinuity region. This issue is a particular bottleneck for the Eulerian SPH as it has compromising potential in the precise simulation of incompressible and compressible flows. Therefore, reducing the overall numerical dissipation especially for the contact discontinuity is one grand challenge to the Eulerian SPH method. For the Riemann-based Eulerian SPH method where the Riemann problem is solved in each interacting particle pair to determine the flux, introducing high-order

Biography: Zhentong Wang (1993-), Male, Ph. D. Candidate,
E-mail: zhentong.wang@tum.de

Corresponding author: Chi Zhang,
E-mail: c.zhang@tum.de

reconstruction is a natural choice. Second-order linear reconstruction such as the monotonic upstream-centered scheme for conservation laws^[19-22] (MUSCL) has been studied, but attempts at higher-order reconstructions have increased substantially to achieve higher accuracy. A one-dimensional fifth-order weighted essentially non-oscillatory (WENO) reconstruction is introduced^[23] and then promoted to multi-dimensional WENO reconstruction at the cost of very low computational efficiency^[24]. Afterwards, a one-dimensional fifth-order WENO reconstruction extending to simulate the multi-dimensional problem in the weakly-compressible SPH (WCSPH) framework is developed with its simple and computational efficiency^[25]. However, this method is only applied to the Lagrangian WCSPH, making its study in the Eulerian SPH for compressible and incompressible flows imperfect and different effects of the WENO reconstruction in the smooth region and contact discontinuity region lacking.

In this paper, we first exploit low-dissipation linearised and Harten-Lax-van Leer contact (HLLC) Riemann solvers for incompressible and compressible flows, respectively. Then, a kernel gradient correction is introduced to achieve first-order consistency and a one-dimensional WENO reconstruction is adopted to improve the overall numerical accuracy further and the shock-capturing performance in the contact discontinuity. A set of benchmark tests for compressible and incompressible flows are studied to investigate the importance of the WENO reconstruction for higher numerical accuracy through comparisons with the theoretical solutions and experimental data. The different effects of the WENO reconstruction on smooth region and contact discontinuity are also explored.

1. SPH methodology

In this section, the governing equations for fluid dynamics in the Eulerian framework are first presented and then the corresponding discretizations by using the Riemann-based Eulerian SPH as well as the kernel correction matrix are given. Furthermore, the low-dissipation Riemann solvers and the WENO reconstruction are presented in detail. Last but not least, the time integration scheme is elaborated.

1.1 Governing equations

The conservation equations for inviscid flow can be written in the Eulerian framework as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \quad (1)$$

where $\mathbf{U}$ and $\mathbf{F}(\mathbf{U})$ are the vector of the con-

served variables and the corresponding fluxes, respectively, given by:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ v(E + p) \end{bmatrix} \quad (2)$$

in two dimension. Here u, v are the components of velocity, ρ is the density and p the pressure. Also, $E = \rho \mathbf{v}^2 / 2 + \rho e$ is the total energy with e denoting the internal energy. For the compressible flows, the system of Eq. (1) is closed by the equation of state (EOS) of

$$p = \rho(\gamma - 1)e \quad (3)$$

where γ is the heat capacity ratio and the speed of sound is derived by

$$c^2 = \frac{\gamma p}{\rho} \quad (4)$$

For the incompressible flows, the EOS is written as

$$p = c^2(\rho - \rho_0) \quad (5)$$

where ρ_0 is the reference density. Following the weakly-compressible assumption for modeling the incompressible flow, we set the sound speed $c = 10U_{\max}$ where $U_{\max}$ denoting the maximum velocity in the flow field to ensure the density variation is less than 1%. To simulate the viscous flow, the viscous force

$$\mathbf{f}^{(v)} = \eta \nabla^2 \mathbf{v} \quad (6)$$

is added to the momentum equation.

1.2 Riemann-based Eulerian SPH method

To discretize the system of Eq. (1), we apply the Riemann-based Eulerian SPH where the flux is determined by solving a one-dimensional Riemann problem constructed along the interacting line of each particle pair, as shown in Fig. 1. The discontinuity locates at the middle point $\bar{\mathbf{r}}_{ij} = (\mathbf{r}_i + \mathbf{r}_j) / 2$, and the left and right states are initially set at particles i and j , respectively, as:

$$(\rho_l, p_l, u_l) = (\rho_i, p_i, \mathbf{v}_i \cdot \mathbf{e}_{ij}),$$

$$(\rho_r, p_r, u_r) = (\rho_j, p_j, \mathbf{v}_j \cdot \mathbf{e}_{ij}) \quad (7)$$

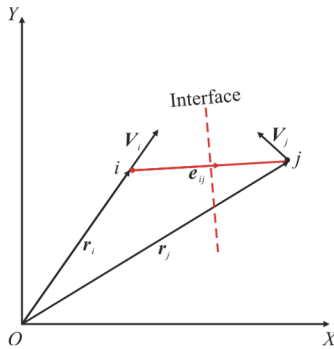


Fig. 1 (Color online) Construction of Riemann problem along the interacting line of particles i and j

Following Ref. [19], Eq. (1) can then be discretized in the Riemann-based Eulerian SPH framework as

$$\begin{aligned} \frac{\partial}{\partial t}(w_i \rho_i) + w_i \sum_j w_j 2(\rho \mathbf{v})_{E,ij}^* \cdot \nabla_i W_{ij} &= 0, \\ \frac{\partial}{\partial t}(w_i \rho_i \mathbf{v}_i) + w_i \sum_j w_j 2[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbf{I}] \cdot \nabla_i W_{ij} &= 0, \\ \frac{\partial}{\partial t}(w_i E_i) + w_i \sum_j w_j 2[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^*] \cdot \nabla_i W_{ij} &= 0 \quad (8) \end{aligned}$$

where w is the volume, $\mathbf{v}$ is the velocity and $\mathbf{I}$ is the identity matrix. Also, $\nabla_i W_{ij} = (\partial W_{ij} / \partial \mathbf{r}_{ij})(\mathbf{r}_{ij} / r_{ij})$ is the gradient of the kernel function with $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and the term $(\cdot)_{E,ij}^*$ is the solution of the particle pair Riemann problem. To achieve first-order consistency, we introduce a kernel correction matrix to compensate for the error caused by particle irregular distribution and kernel truncation. Following Ref. [26], the kernel correction matrix $\mathbf{B}_i$ reads

$$\mathbf{B}_i = [\sum_j (\mathbf{r}_j - \mathbf{r}_i) \otimes \nabla_i W_{ij} w_j]^{-1} \quad (9)$$

Then, we rewrite the kernel gradient as

$$\tilde{\nabla}_i W_{ij} = \frac{1}{2}(\mathbf{B}_i + \mathbf{B}_j) \nabla_i W_{ij} \quad (10)$$

to ensure the conservation property of the SPH discretization

1.3 Riemann solver with dissipation limiter

For the incompressible flows, we use the linearised Riemann solver and assume that the intermediate states satisfy $p_l^* = p_r^* = p^*$ and $u_l^* = u_r^* = u^*$, as shown in Fig. 2. Following Ref. [27], the low-dissipation Riemann solver reads

$$\begin{aligned} u^* &= \frac{1}{2}(u_l + u_r) + \frac{1}{2} \frac{p_l - p_r}{\bar{\rho} \bar{c}_0} \beta_{\text{linearisd}}^2, \\ p^* &= \frac{1}{2}(p_l + p_r) + \frac{1}{2} \bar{\rho} \bar{c} \beta_{\text{linearisd}} (u_l - u_r) \end{aligned} \quad (11)$$

with $\bar{c}$ denoting the average sound speed. Note that the dissipation limiter $\beta_{\text{linearisd}}^{[16]}$ is defined as

$$\beta_{\text{linearisd}} = \min \left(\eta_{\text{linearisd}} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right) \quad (12)$$

with the constant parameter $\eta_{\text{linearisd}} = 15$.

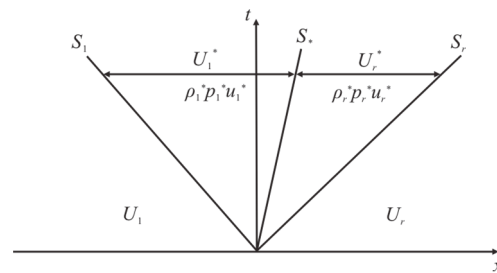


Fig. 2 Simplified Riemann fan with two intermediate states

For the compressible flows, we implement the HLLC Riemann solver^[28-31] which takes the advantage of capturing shock waves. In the HLLC Riemann solver, the smallest and the largest wave speed S_l and S_r characterize the expansion and compression wave, respectively. The contact discontinuity with wave speed S_* separates the middle state into (ρ_l^*, p_l^*, u_l^*) and (ρ_r^*, p_r^*, u_r^*) . Then, the solutions in the star region are given by:

$$\begin{aligned} u^* &= \frac{\rho_l u_l c_l + \rho_r u_r c_r}{\rho_l c_l + \rho_r c_r} + \frac{p_l - p_r}{\rho_l c_l + \rho_r c_r}, \\ p^* &= \frac{p_l + p_r}{2} + \frac{1}{2} [\rho_r c_r (u^* - u_r) - \rho_l c_l (u_l - u^*)], \end{aligned}$$

$$\rho_{l/r}^* = \rho_{l/r} \frac{S_{l/r} - q_{l/r}}{S_{l/r} - u^*},$$

$$E_{l/r}^* = \frac{(S_{l/r} - q_{l/r})E_{l/r} - p_{l/r}q_{l/r} + p^*u^*}{S_{l/r} - u^*} \quad (13)$$

where $q_{l/r}$ is the magnitude of the velocity. Note that a dissipation limiter is also applied in the HLLC Riemann solver. With the star states in hand, the corresponding flux of the HLLC solution to the Riemann problem can be expressed as:

$$\mathbf{F}_{\text{HLLC}} = \mathbf{F}, \quad 0 \leq s_l \quad (14a)$$

$$\mathbf{F}_{\text{HLLC}} = \mathbf{F}_L^*, \quad s_l \leq 0 \leq s_* \quad (14b)$$

$$\mathbf{F}_{\text{HLLC}} = \mathbf{F}_R^*, \quad s_* \leq 0 \leq s_r \quad (14c)$$

$$\mathbf{F}_{\text{HLLC}} = \mathbf{F}_R, \quad 0 \geq s_r \quad (14d)$$

1.4 WENO reconstruction

Following Ref. [25], the 4-point stencil with the incremental size is constructed along each interacting particle pair as shown in Fig. 3. For each stencil, the variable of the mid-point is defined as

$$q_{1/2} = \sum_{k=1}^3 w_k q_{1/2}^{(k)} \quad (15)$$

where

$$\begin{aligned} q_{1/2}^{(1)} &= \frac{1}{2}q_0 + \frac{1}{2}q_1, \quad q_{1/2}^{(2)} = -\frac{1}{2}q_{-1} + \frac{3}{2}q_0, \\ q_{1/2}^{(3)} &= \frac{1}{3}q_0 + \frac{5}{6}q_1 - \frac{1}{6}q_2 \end{aligned} \quad (16)$$

are the candidate stencils and w_k represent the corresponding non-linear weights calculated as

$$w_k = \frac{\alpha_k}{\sum_{s=1}^3 \alpha_s} \quad (17)$$

with

$$\begin{aligned} \alpha_1 &= d_1 \left(C + \frac{\tau_4}{\beta_1 + \varepsilon} \frac{\tau_4}{\beta_{12} + \varepsilon} \right), \\ \alpha_2 &= d_2 \left(C + \frac{\tau_4}{\beta_2 + \varepsilon} \frac{\tau_4}{\beta_{12} + \varepsilon} \right), \end{aligned}$$

$$\alpha_3 = d_3 \left(C + \frac{\tau_4}{\beta_3 + \varepsilon} \right) \quad (18)$$

where $d_1 = 1/3$, $d_2 = 1/6$, $d_3 = 1/2$, $\varepsilon = 10^{-6}$ and C is a physically-motivated scale-separation parameter. Besides, a global reference smoothness indicator τ_4 given by

$$\begin{aligned} \tau_4 &= [q_{-1}(547q_{-1} - 2522q_0 + 1922q_1 - 494q_2) + \\ & q_0(3443q_0 - 5966q_1 + 1602q_2) + \\ & q_1(2843q_1 - 1642q_2) + 267q_2q_2] / 240 \end{aligned} \quad (19)$$

is introduced to improve the overall accuracy^[32-33]. Note that the smoothness indicators in Eq. (18) are given as:

$$\begin{aligned} \beta_1 &= (q_1 - q_0)^2, \quad \beta_2 = (q_0 - q_{-1})^2, \\ \beta_{12} &= \frac{1}{4}(q_{-1} - q_1)^2 + \frac{13}{12}(q_{-1} - 2q_0 + q_1)^2, \\ \beta_3 &= \frac{13}{12}(q_0 - 2q_1 + q_2)^2 + \frac{1}{4}(3q_0 - 4q_1 + q_2)^2 \end{aligned} \quad (20)$$

For each interacting particle pair i and j , the variable values at the stencil point are defined as:

$$\begin{aligned} q_{-1} &= \Phi_i - \nabla \Phi_i \cdot \mathbf{r}_{ij}, \quad q_0 = \Phi_i, \quad q_1 = \Phi_j, \\ q_2 &= \Phi_j + \nabla \Phi_j \cdot \mathbf{r}_{ij} \end{aligned} \quad (21)$$

where

$$\nabla \Phi_i = \sum_j \frac{m_j}{\rho_j} (\Phi_j - \Phi_i) \nabla_i W_{ij} \quad (22)$$

is the gradient of primitive value Φ_j with a standard SPH approximation. Afterwards, the left state of the Riemann problem is $\Phi_L = q_{1/2}$ and the corresponding right state Φ_R can be calculated from Eq. (15) with the mirrored stencil defined as:

$$\begin{aligned} q_{-1} &= \Phi_j + \nabla \Phi_j \cdot \mathbf{r}_{ij}, \quad q_0 = \Phi_j, \quad q_1 = \Phi_i, \\ q_2 &= \Phi_i - \nabla \Phi_i \cdot \mathbf{r}_{ij} \end{aligned} \quad (23)$$

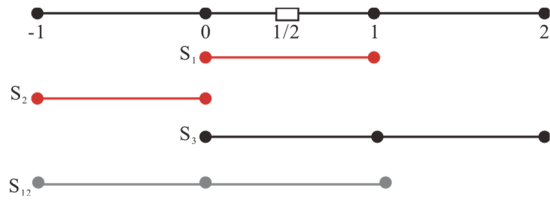


Fig. 3 (Color online) Four-point stencil and candidate stencils for the incremental-stencil WENO reconstruction

1.5 Time integration

For the time integration, we apply the verlet scheme^[34] to achieve 2nd-order accuracy. We denote the value at the beginning of a time-step by superscript n , the mid-point by $n+1/2$ and the end by $n+1$. At first, total energy and density are updated to the mid-point by:

$$E_i^{n+1/2} = E_i^n + \frac{1}{2} \delta t \left(\frac{dE_i}{dt} \right)^n, \quad \rho_i^{n+1/2} = \rho_i^n + \frac{1}{2} \delta t \left(\frac{d\rho_i}{dt} \right)^n \quad (24)$$

Accordingly, the pressure and internal energy are also calculated. Then, the momentum flux at the new time step is updated as

$$(\rho v)_i^{n+1} = (\rho v)_i^n + \delta t \left[\frac{d(\rho v)_i}{dt} \right]^{n+1/2} \quad (25)$$

Afterwards, mass and energy change rates are calculated. Finally, the energy and density at the new step are updated as:

$$E_i^{n+1} = E_i^{n+1/2} + \frac{1}{2} \delta t \left(\frac{dE_i}{dt} \right)^{n+1},$$

$$\rho_i^{n+1} = \rho_i^{n+1/2} + \frac{1}{2} \delta t \left(\frac{d\rho_i}{dt} \right)^{n+1} \quad (26)$$

For numerical stability, the time step size is obtained by

$$\Delta t_{ac} = \text{CFL} \left(\frac{h}{c_0 + U_{\max}} \right) \quad (27)$$

with $U_{\max}$ denoting the maximum flow speed and $\text{CFL} = 0.6/d$, where d is the dimension.

2. Numerical results

In this part, a set of numerical examples for com-

pressible and incompressible flows are studied to investigate the accuracy, robustness and versatility of the proposed Riemann-based Eulerian SPH method. It is worth noting that the linearised and HLLC Riemann solvers are for the incompressible and compressible flows, respectively. The Wenland C2^[35] kernel function with a smoothing length $h = 1.3d_p$, d_p denoting the initial particle spacing is applied. In the subsequent numerical examples, with the application of the correction matrix and the dissipation limiter through all the examples, the Eulerian SPH with the Riemann solver is denoted as “ESPH” and the Eulerian SPH with the Riemann solver and WENO reconstruction is denoted by “ESPH-WENO”, for clarity.

2.1 Shock-tube problems

Following Refs. [36–37], we simulate two shock-tube cases including the Sod and Lax problems where the heat capacity ratio is $\gamma = 1.4$ and the domain size is $x \in [0, 1]$. For the convergence study, three different spatial resolutions of $d_p = 1/100$, $1/200$ and $1/400$ are investigated. The initial condition of the Sod problem is:

$$(\rho, u, v, p) = (1, 0, 0, 1), \quad 0 \leq x < 0.5 \quad (28a)$$

$$(\rho, u, v, p) = (0.125, 0, 0, 0.1), \quad 0.5 \leq x < 1.0 \quad (28b)$$

and the final time is $t = 0.2$. The initial condition for the Lax problem is:

$$(\rho, u, v, p) = (0.445, 0.698, 0, 3.582), \quad 0 \leq x < 0.5 \quad (29a)$$

$$(\rho, u, v, p) = (0.5, 0, 0, 0.5710), \quad 0.5 \leq x < 1.0 \quad (29b)$$

and its final time is $t = 0.14$.

Figures 4–6 show the present density and velocity profiles of the numerical results for both Sod and Lax problems and the comparisons against the analytical solution. With the spatial resolution of $1/100$, the ESPH-WENO exhibits improved accuracy by significantly reducing the numerical dissipation. More importantly, the dissipation in the contact discontinuity region is more reduced by the present WENO reconstruction than those in the continuous expansion and the shock region. This is due to the fact that the WENO reconstruction with the 4-point stencil gives a more accurate and physically-driven solution in the contact discontinuity region. With the increase of the spatial resolution, the result is rapidly convergent to the theoretical solution.

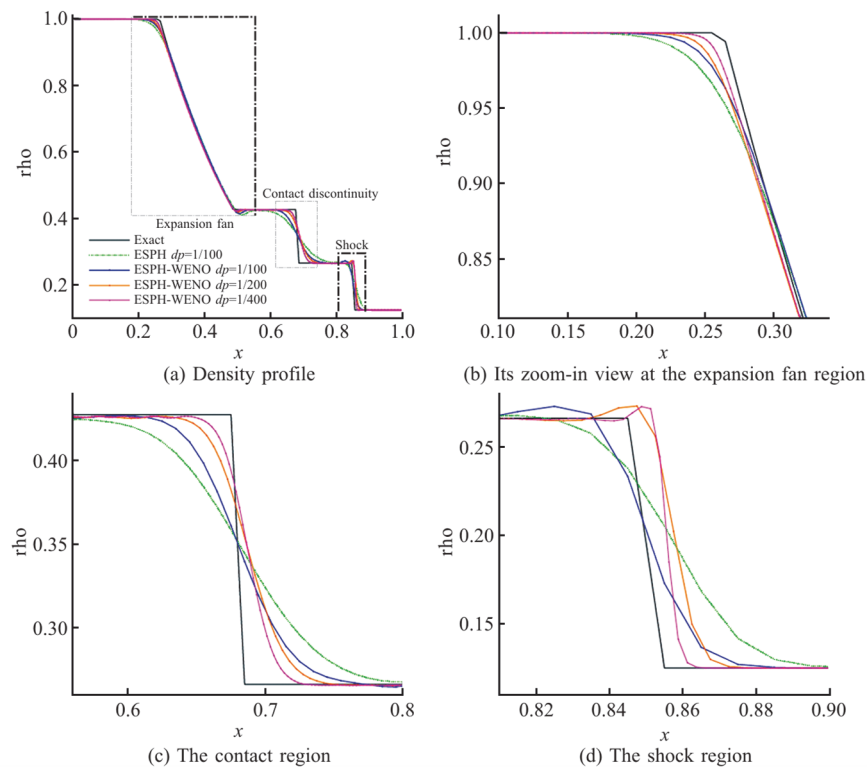


Fig. 4 (Color online) The Sod problem: The present results and their comparisons against the analytical solution

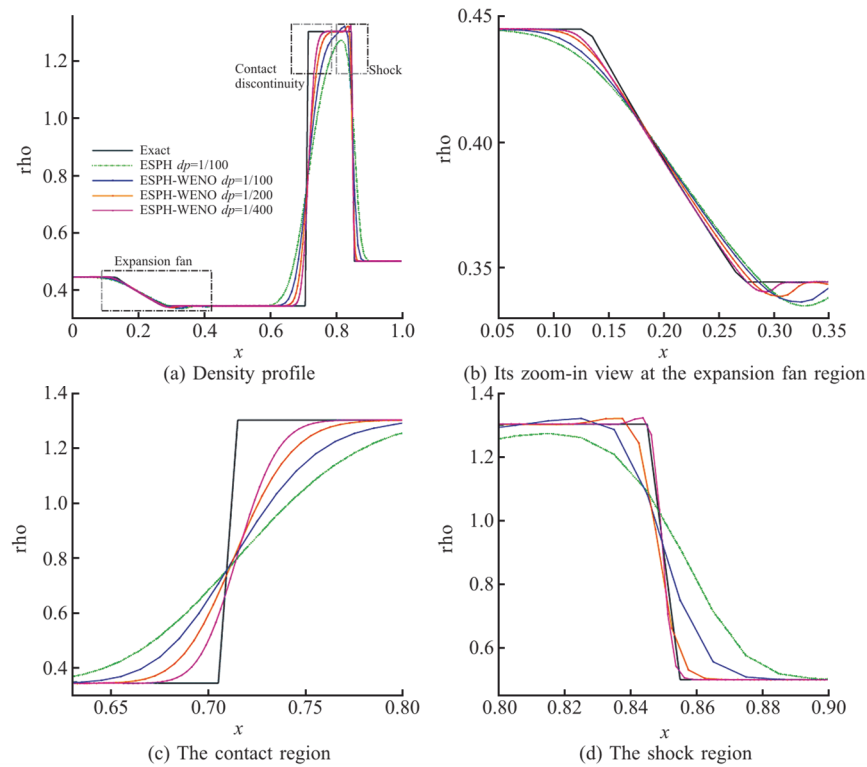


Fig. 5 (Color online) The Lax problem: The present results and their comparisons against the analytical solution

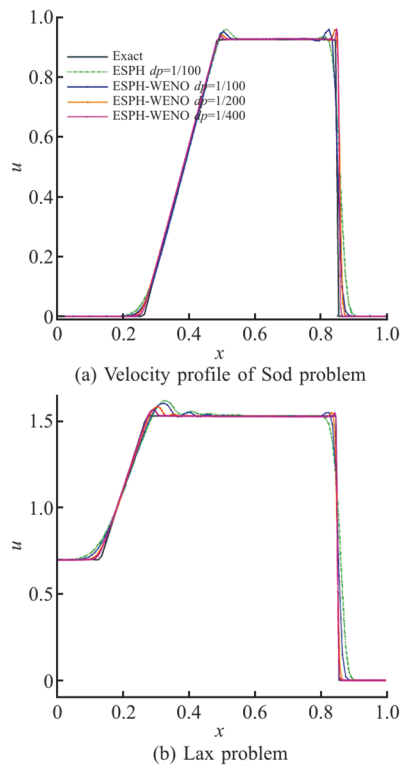


Fig. 6 (Color online) Shock-tube problem

2.2 The Shu-Osher problem

In this section, the Shu-Osher problem of a Mach 3 shock wave interacting with a sine entropy wave is studied to validate the accuracy and robustness of the proposed ESPH-WENO method. The domain is $x \in [0, 10]$ and the initial condition reads:

$$(\rho, u, v, p) = (3.857, 2.629, 0, 10.333), \quad x \leq 1 \quad (30a)$$

$$(\rho, u, v, p) = (0.5, 0, 0, 0.5710) \quad \text{otherwise} \quad (30b)$$

and the final time is $t = 1.8$.

Figure 7 shows the density profile of the present results and their comparisons with reference data obtained by a high-resolution scheme^[38]. The overall density distribution is well captured by the ESPH method with or without the WENO reconstruction. Compared with the result of the original ESPH method, the ESPH-WENO exhibits a much sharper density profile and less dissipation, implying that the WENO reconstruction effectively improves the numerical accuracy especially at the contact region where the limiter has little effect. Also, the convergence study shows that the result is rapidly convergent with the increase of spatial resolution.

2.3 Double Mach reflection of a strong shock

In this part, we consider a two-dimensional double Mach reflection of a strong shock to demonstrate

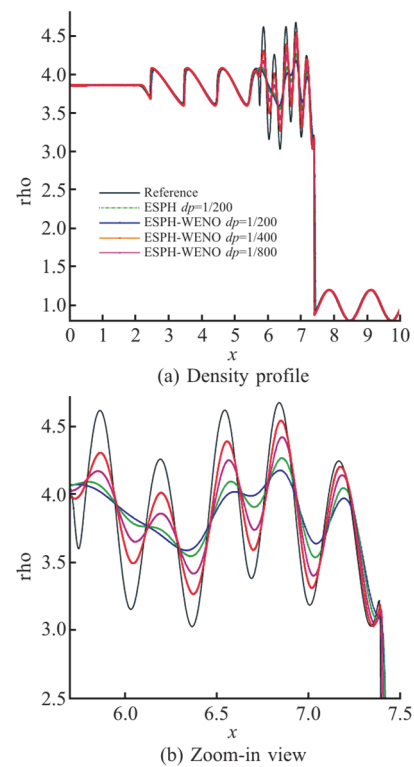


Fig. 7 (Color online) Shu-Osher problem

the robustness of the proposed ESPH-WENO method. Following Woodward^[39], the initial condition is:

$$(\rho, u, v, p) = (1.4, 0, 0, 1), \quad y \leq 1.732(x - 0.1667) \quad (31a)$$

$$(\rho, u, v, p) = (8, 7.15, -4.13, 116.83) \quad \text{otherwise} \quad (31b)$$

In this study, the domain size is $(x, y) \in [0, 4] \times [0, 1]$ and the final time is $t = 2$. Initially, a right-moving Mach 10 shock wave is activated at $(x, y) = (0, 1)$ with a sixty-degree angle along the x-axis. The bottom and left-hand boundary are the reflection wall boundary from $x = 1/6$ to $x = 4$ and the post-shock condition, respectively. The top boundary is under exact motion condition and the right boundary at $x = 4$ is zero-gradient condition.

Figure 8 portrays the shock propagation with the density contour obtained by the ESPH and the ESPH-WENO methods. It can be observed that the Mach stem and the near-wall jet flow features are well captured in both methods with a spatial resolution of $1/240$. Without the WENO reconstruction, the curve of the near-wall jet is very flat, as shown in Fig. 8(a), indicating that there is relatively large numerical dissipation. By introducing the WENO reconstruction, the curve of the near-wall jet becomes much sharper, as shown in Fig. 8(b). These results demonstrate that

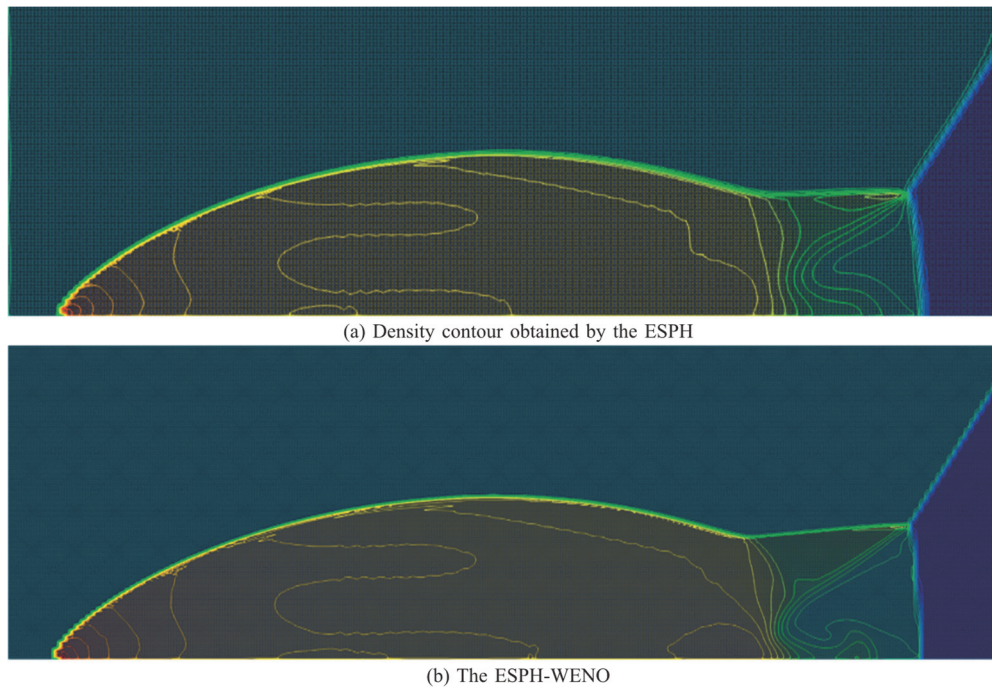


Fig. 8 (Color online) Double Mach reflection of a strong shock

using only the dissipation limiter to reduce the dissipation is not enough to capture the strong shock wave propagation which is well captured by introducing the WENO reconstruction.

2.4 Taylor-Green vortex flow

In this section, we consider an example of incompressible flow, viz. Taylor-Green vortex flow, to investigate the performance of the proposed ESPH-WENO method in the simulation of incompressible flow. With the periodic boundary in x - and y -directions, the velocity field reads

$$\begin{aligned} u(x, y, t) &= -\exp^{bt} \cos(2\pi x) \sin(2\pi y), \\ v(x, y, t) &= \exp^{bt} \sin(2\pi x) \cos(2\pi y) \end{aligned} \quad (32)$$

where $b = 8\pi^2 / Re$ is the decay rate with Re the Reynolds number. The initial velocity distribution is defined at $t = 0$. Note that the corresponding total kinetic energy decay rate is $-16\pi^2 / Re$. Also, the domain size is $(x, y) \in [0, 1] \times [0, 1]$ and three different resolutions $d_p = 1/25$, $1/50$ and $1/100$ are applied for the convergence study.

Figure 9 presents the time decay of the kinetic energy obtained by the ESPH and the ESPH-WENO methods and their comparisons with the analytical solution. With the WENO reconstruction, the ESPH-WENO exhibits improved accuracy by decreasing

numerical dissipation. While the ESPH shows 1st-order convergence, the ESPH-WENO achieves approximate 2nd-order convergence. It can be concluded that the introduction of the WENO reconstruction is not only to greatly reduce the numerical dissipation but also to raise the convergence order of the ESPH method.

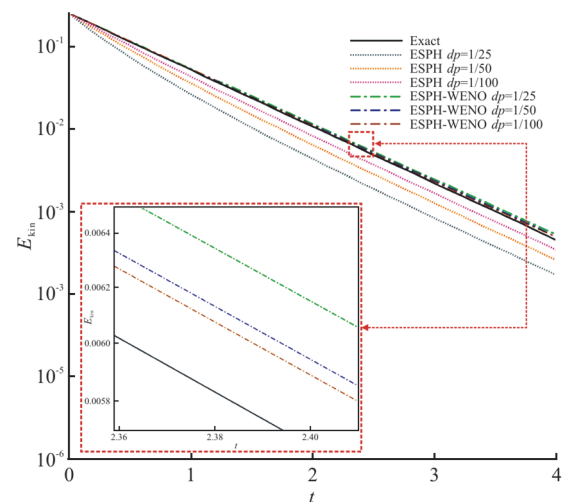


Fig. 9 (Color online) Taylor-Green vortex flow ($Re = 100$): Decay of the kinetic energy obtained by the ESPH and the ESPH-WENO methods and the comparisons against analytical solution

2.5 Double shear-layer flow

In this part, we consider an inviscid incom-

pressible flow of double shear-layer to demonstrate the low-dissipation feature of the proposed ESPH-WENO method. For this test, the domain size is a square with unit length and all boundaries are set as periodic boundary conditions. The initial moment of the double shear-layer flow is represented on a shear layer of finite thickness giving a perturbation with a small vertical velocity, and the velocity field is:

$$u_x = \tanh\left(\frac{y-0.25}{\rho}\right) \quad \text{for } y \leq 0.5 \quad (33a)$$

$$u_x = \tanh\left(\frac{0.75-y}{\rho}\right) \quad \text{for } y > 0.5 \quad (33b)$$

$$u_y = \delta \sin(2\pi x) \quad (33c)$$

where $\rho = 1/30$, $\delta = 0.05$. To discretize the system, the spatial resolution is set as $d_p = 1/64$.

Figure 10 presents the vortex contours at different time instants obtained by the ESPH and the ESPH-WENO methods. As expected, the ESPH method exhibits excessive numerical dissipation and fails to correctly capture the vortex structure. On the other hand, the ESPH-WENO method reasonably captures the vortex structure, implying the numerical accuracy is significantly improved with the WENO reconstruction.

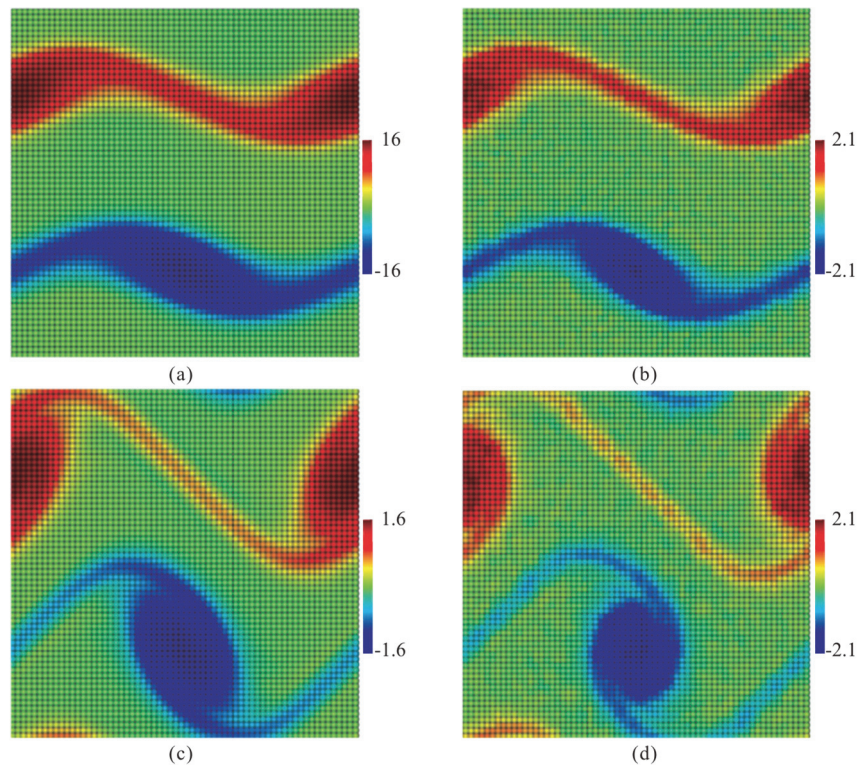


Fig. 10 (Color online) Flow of double shear layer: The vorticity contour ranging from -16 to 16 obtained by the ESPH at $t = 0.8$ (a), The vorticity contour ranging from -21 to 21 by the ESPH-WENO at $t = 0.8$ (b), The vorticity contour ranging from -16 to 16 by the ESPH at $t = 1.2$ (c) and The vorticity contour ranging from -21 to 21 by the ESPH-WENO at $t = 1.2$ (d)

2.6 Lid-driven cavity flow

Here, we consider an incompressible flow of a square lid-driven cavity with a length of L where all boundaries are wall boundary conditions. The upper wall is moving with a given velocity U_{wall} along the x -direction, as shown in Fig. 11, to initialize the cavity. Two different Reynolds numbers including $Re = 400$, $Re = 1000$ are considered to validate the proposed method by comparing with the results of Ghia^[40].

Figure 12 portrays the velocity profile obtained by the present methods and their comparisons with the reference data reported by Ghia^[40]. Compared with the results of ESPH, those obtained by the ESPH-WENO show improved accuracy and exhibit a low-dissipation feature. It is worth noting that the present ESPH-WENO achieves approximate 2nd-order accuracy as shown in the convergence study in Fig. 12.

2.7 Flow around a circular cylinder

To demonstrate the versatility of the proposed ESPH-WENO method, we consider a FSI problem of flow around a cylinder. In this case, the drag and lift coefficients can be derived by:

$$C_D = \frac{2F_D}{\rho_\infty u_\infty^2 A}, \quad C_L = \frac{2F_L}{\rho_\infty u_\infty^2 A} \quad (34)$$

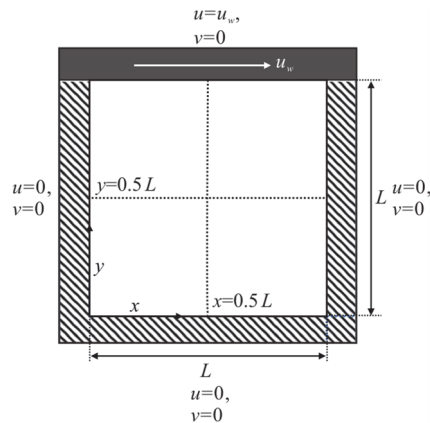


Fig. 11 Lid-driven cavity flow: Geometry and boundary conditions

where F_D is the drag force, F_L the lift force and A the contact area. Note that the Reynolds number $Re = u_\infty \rho_\infty D / \mu$ is set as 100 with velocity $u_\infty = 1$ and density $\rho_\infty = 1$ as well as the cylinder diameter $D = 2$. Also, the Strouhal number Sr is defined as fD / u_∞ with f the vortex shedding frequency. We consider in this study the domain size as $[12.5D, 7.5D]$ where the centre of a cylinder is located at

$[7.5D, 7.5D]$ and the boundary condition is defined as the far-field boundary condition.

Figure 13 shows the flow field with the velocity contour at dimensionless time $t = 300$ obtained by the ESPH (a) and the ESPH-WENO (b) methods. It is easy to observe that the vortex structure is over-dissipated by the ESPH method, while reasonably captured by the ESPH-WENO method. For quantitative validation, Table 1 presents the drag and lift coefficients obtained by the present simulation and the data in the literature^[41–44]. In general, a good agreement with the reference data is noted. It can be concluded that the proposed ESPH-WENO paves the way for the SPH simulation of complex FSI applications in engineering within the Eulerian framework.

3. Conclusion

In this paper, we introduce a WENO reconstruction to further improve the accuracy of the ESPH method. The underlying principle is to obtain a high-accuracy WENO reconstruction of the Riemann states which determines the flow flux by introducing a one-dimensional 4-point stencil along each interacting particle pair. A set of numerical examples for the compressible and incompressible flows are studied to investigate the accuracy, robustness and versatility of the proposed method.

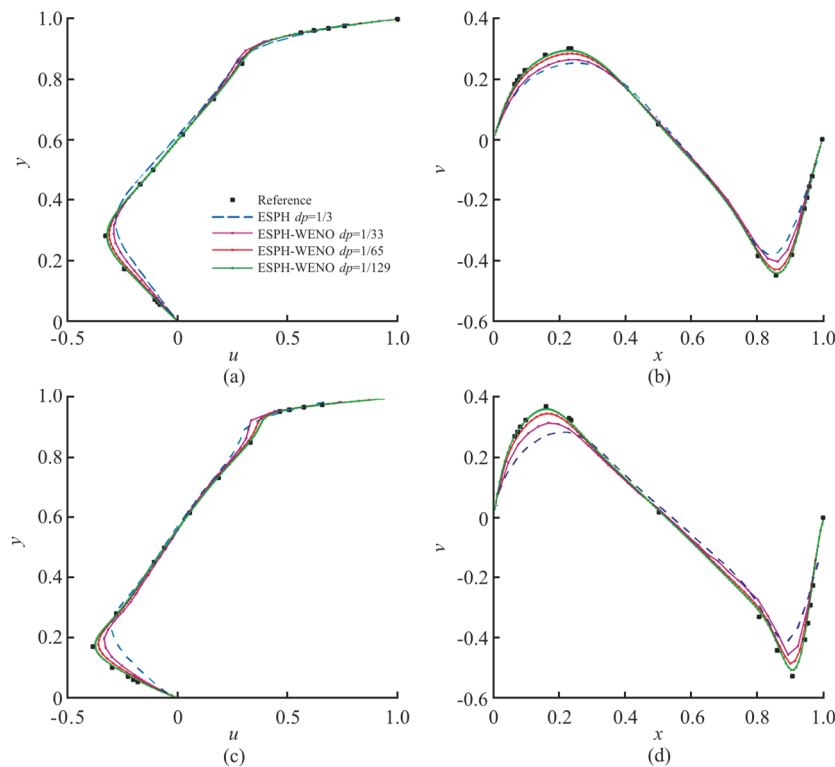


Fig. 12 (Color online) Lid-driven cavity flow: The horizontal velocity component u along $x = 0.5L$ for $Re = 400$ (a), The vertical velocity component v along $y = 0.5L$ for $Re = 400$ (b), The horizontal velocity component u along $x = 0.5L$ for $Re = 1000$ (c) and The vertical velocity component v along $y = 0.5L$ for $Re = 1000$ (d)

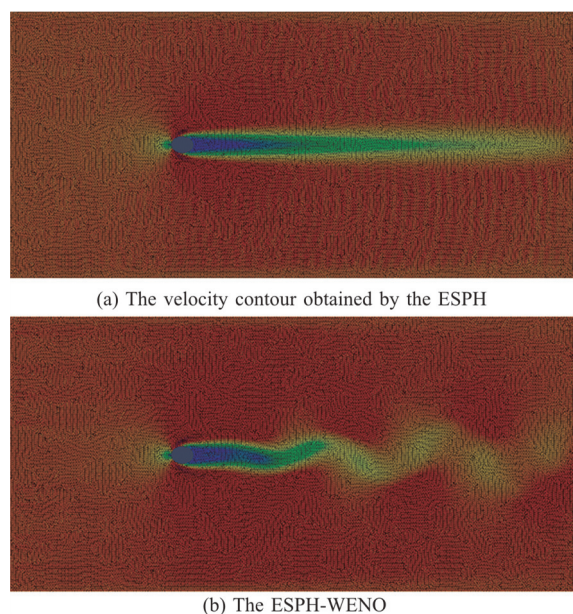


Fig. 13 (Color online) Flow around a cylinder with $Re = 100$ at dimensionless time $t = 300$

Table 1 The drag and lift coefficients in the literature

Parameters	C_D	C_L	S_t
White ^[41]	1.46	-	-
Chiu et al. ^[42]	1.35 ± 0.012	± 0.303	0.166
Le et al. ^[43]	1.37 ± 0.009	± 0.323	0.160
Liu et al. ^[44]	1.35 ± 0.012	± 0.339	0.165
ESPH	1.46 ± 0.010	± 0.320	0.171
ESPH-WENO	1.43 ± 0.011	± 0.360	0.174

Acknowledgement

This work was supported by the China Scholarship Council (Grant No. No.201906120035), Chi Zhang and Xiangyu Hu would like to express their gratitude to Deutsche Forschungsgemeinschaft (DFG) for their sponsorship (Grant No. DFG HU1527/12-4).

Compliance with ethical standards

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent: Informed consent was obtained from all individual participants included in the study.

References

- [1] Lucy L. B. A numerical approach to the testing of the fission hypothesis [J]. *Astronomical Journal*, 1977, 82:

1013-1024.

- [2] Gingold R. A., Monaghan J. J. Smoothed particle hydrodynamics: theory and application to non-spherical stars [J]. *Monthly Notices of the Royal Astronomical Society*, 1977, 181(3): 375-389.
- [3] Monaghan J. J. Simulating free surface flows with SPH [J]. *Journal of Computational Physics*, 1994, 110(2): 399-406.
- [4] Libersky L. D., Petschek A. G. Smooth particle hydrodynamics with strength of materials (Advances in the free-Lagrange method including contributions on adaptive gridding and the smooth particle hydrodynamics method) [M]. Berlin, Germany: Springer, 1991, 248-257.
- [5] Gómez-Gesteira M., Dalrymple R. A. Using a three-dimensional smoothed particle hydrodynamics method for wave impact on a tall structure [J]. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 2004, 130(2): 63-69.
- [6] Mokos A., Rogers B. D., Stansby P. K. et al. Multi-phase SPH modelling of violent hydrodynamics on GPUs [J]. *Computer Physics Communications*, 2015, 196: 304-316.
- [7] Longshaw S. M., Rogers B. D. Automotive fuel cell sloshing under temporally and spatially varying high acceleration using GPU-based smoothed particle hydrodynamics (SPH) [J]. *Advances in Engineering Software*, 2015, 83: 31-44.
- [8] Luo M., Khayyer A., Lin P. Particle methods in ocean and coastal engineering [J]. *Applied Ocean Research*, 2021, 114: 102734.
- [9] Zhang C., Zhu Y. J., Wu D. et al. Smoothed particle hydrodynamics: Methodology development and recent achievement [J]. *Journal of Hydrodynamics*, 2022, 34(5): 767-805.
- [10] Lind S. J., Stansby P. High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion [J]. *Journal of Computational Physics*, 2016, 326: 290-311.
- [11] Noutcheuwa R. K., Owens R. G. A new incompressible smoothed particle hydrodynamics-immersed boundary method [J]. *International Journal of Numerical and Analytical Modeling Series B*, 2012, 3(2): 126-167.
- [12] Lind S., Stansby P. Investigations into higherorder incompressible SPH [C]. *Proceedings 10th SPHERIC International Workshop*, Parma, Italy, 2015.
- [13] Nasar A. M. Eulerian and Lagrangian smoothed particle hydrodynamics as models for the interaction of fluids and flexible structures in biomedical flows [R]. Manchester, UK: The University of Manchester, 2016.
- [14] Ferrari A., Dumbser M., Toro E. F. et al. A new 3D parallel SPH scheme for free surface flows [J]. *Computers and Fluids*, 2009, 38(6): 1203-1217.
- [15] Molteni D., Colagrossi A. A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH [J]. *Computer Physics Communications*, 2009, 180(6): 861-872.
- [16] Zhang C., Hu X., Adams N. A. A weakly compressible SPH method based on a low-dissipation Riemann solver [J]. *Journal of Computational Physics*, 2017, 335: 605-620.
- [17] Bonet J., Lok T. S. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations [J]. *Computer Methods in Applied Mechanics and Engineering*, 1999, 180(1-2): 97-115.
- [18] Zago V., Schulze L. J., Bilotta G. et al. Overcoming excessive numerical dissipation in SPH modeling of water waves [J]. *Coastal Engineering*, 2021, 170: 104018.
- [19] Vila J. P. On particle weighted methods and smooth

- particle hydrodynamics [J]. *Mathematical Models and Methods in Applied Sciences*, 1999, 9(2): 161-209.
- [20] Inutsuka S. I. Reformulation of smoothed particle hydrodynamics with Riemann solver [J]. *Journal of Computational Physics*, 2002, 179(1): 238-267.
- [21] Iwasaki K., Inutsuka S. I. Smoothed particle magnetohydrodynamics with a Riemann solver and the method of characteristics [J]. *Monthly Notices of the Royal Astronomical Society*, 2011, 418(3): 1668-1688.
- [22] Murante G., Borgani S., Brunino R. et al. Hydrodynamic simulations with the Godunov smoothed particle hydrodynamics [J]. *Monthly Notices of the Royal Astronomical Society*, 2011, 417(1): 136-153.
- [23] Zhang X., Tian H., Kuo L. et al. A contact SPH method with high-order limiters for simulation of inviscid compressible flows [J]. *Communications in Computational Physics*, 2013, 14(2): 425-442.
- [24] Avesani D., Dumbser M., Bellin A. A new class of moving-least-squares WENO-SPH schemes [J]. *Journal of Computational Physics*, 2014, 270: 278-299.
- [25] Zhang C., Xiang G., Wang B. et al. A weakly compressible SPH method with WENO reconstruction [J]. *Journal of Computational Physics*, 2019, 392: 1-18.
- [26] Randles P., Libersky L. D. Smoothed particle hydrodynamics: some recent improvements and applications [J]. *Computer Methods in Applied Mechanics and Engineering*, 1996, 139(1-4): 375-408.
- [27] Toro E. F. Riemann solvers and numerical methods for fluid dynamics: A practical introduction [M]. Berlin, Germany: Springer Science and Business Media, 2013.
- [28] Harten A., Lax P. D., Leer B. V. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws [J]. *SIAM Review*, 1983, 25(1): 35-61.
- [29] Toro E. F., Spruce M., Speares W. Restoration of the contact surface in the HLL-Riemann solver [J]. *Shock Waves*, 1994, 4: 25-34.
- [30] Hu X., Adams N. A., Iaccarino G. On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow [J]. *Journal of Computational Physics*, 2009, 228(17): 6572-6589.
- [31] Batten P., Clarke N., Lambert C. et al. On the choice of wavespeeds for the HLLC Riemann solver [J]. *SIAM Journal on Scientific Computing*, 1997, 18(6): 1553-1570.
- [32] Fu L., Hu X. Y., Adams N. A. A family of high-order targeted ENO schemes for compressible-fluid simulations [J]. *Journal of Computational Physics*, 2016, 305: 333-359.
- [33] Tritschler V. K., Hu X. Y., Hickel S. et al. Numerical simulation of a Richtmyer-Meshkov instability with an adaptive central-upwind sixth-order WENO scheme [J]. *Physica Scripta*, 2013, T155: 014016.
- [34] Zhang C., Rezavand M., Hu X. A multi-resolution SPH method for fluid-structure interactions [J]. *Journal of Computational Physics*, 2021, 429: 110028.
- [35] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree [J]. *Advances in Computational Mathematics*, 1995, 4: 389-396.
- [36] Sod G. A. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws [J]. *Journal of Computational Physics*, 1978, 27(1): 1-31.
- [37] Lax P. D. Weak solutions of nonlinear hyperbolic equations and their numerical computation [J]. *Communications on Pure and Applied Mathematics*, 1954, 7(1): 159-193.
- [38] Zhu Y., Hu X. An L2-norm regularized incremental-stencil WENO scheme for compressible flows [J]. *Computers and Fluids*, 2020, 213: 104721.
- [39] Woodward P., Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks [J]. *Journal of Computational Physics*, 1984, 54(1): 115-173.
- [40] Ghia U., Ghia K. N., Shin C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method [J]. *Journal of Computational Physics*, 1982, 48(3): 387-411.
- [41] White F. M., Majdalani J. Viscous fluid flow [M]. New York, USA: McGraw-Hill, 2006.
- [42] Chiu P. H., Lin R. K., Sheu T. W. A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-Stokes equations in time-varying complex geometries [J]. *Journal of Computational Physics*, 2010, 229(12): 4476-4500.
- [43] Le D. V., Khoo B. C., Peraire J. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries [J]. *Journal of Computational Physics*, 2006, 220(1): 109-138.
- [44] Liu C., Zheng X., Sung C. Preconditioned multigrid methods for unsteady incompressible flows [J]. *Journal of Computational Physics*, 1998, 139(1): 35-57.

A.4 Paper IV

Zhentong Wang, Bo Zhang, Oskar J. Haidn, Xiangyu Hu

A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids

In *Journal of Computational Physics*, Volume 519, 2024, p. 113385, DOI: <https://doi.org/10.1016/j.jcp.2024.113385>.

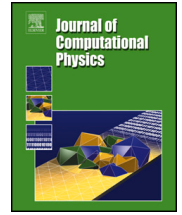
Copyright © 2024 Elsevier. Reprinted with permission.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.



Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A fourth-order kernel for improving numerical accuracy and stability in Eulerian SPH for fluids and total Lagrangian SPH for solids

Zhentong Wang, Bo Zhang, Oskar J. Haidn, Xiangyu Hu *

TUM School of Engineering and Design, Technical University of Munich, Garching, 85748, Germany

ARTICLE INFO

Dataset link: <https://www.sphinxsys.org>Dataset link: <https://github.com/Xiangyu-Hu/SPHinxsys>

Keywords:

Truncated Laguerre-Gauss kernel

Fourth-order

Numerical accuracy

Stability

Computational efficiency

Eulerian SPH

Total Lagrangian SPH

ABSTRACT

The error of smoothed particle hydrodynamics (SPH) using a kernel for particle-based approximation mainly arises from smoothing and integration errors. The choice of the kernel significantly impacts numerical accuracy, stability and computational efficiency. Currently, the most popular kernels, such as B-spline, truncated Gaussian (for compact support), and Wendland kernels, have 2nd-order smoothing errors, and the Wendland kernel has become mainstream in the SPH community due to its stability and accuracy. Since the particle distribution after relaxation can achieve fast convergence of integration error with respect to support radius, it is logical to choose kernels with higher-order smoothing error to improve the numerical accuracy. In this paper, the error of the 4th-order Laguerre-Wendland kernel proposed by Litvinov et al. [1] is revisited, and another 4th-order truncated Laguerre-Gauss kernel is further analyzed and considered to replace the widely used Wendland kernel. The proposed kernel has the following three properties: One is that it avoids the pair-instability problem during the relaxation process, unlike the original truncated Gaussian kernel, and achieves much less relaxation residue than the Wendland and Laguerre-Wendland kernels; One is the truncated compact support size is the same as the non-truncated one of the Wendland kernel, which leads to both kernels' comparably computational efficiency; Another is that the truncation error of this kernel is much less than that of the Wendland kernel. Furthermore, a comprehensive set of 2D and 3D benchmark cases on Eulerian SPH for fluid dynamics and total Lagrangian SPH for solid dynamics validate the considerably improved numerical accuracy of the truncated Laguerre-Gauss kernel without introducing extra computational effort.

1. Introduction

Smoothed particle hydrodynamics (SPH) is a meshless method initially proposed by Lucy [2], Gingold and Monaghan [3] and has been widely used in fluid dynamics [3], solid mechanics [4], and other scientific and industrial applications [5–8]. Since the particle-based approximation in SPH is formulated with a Gaussian-like smoothing kernel function with compact support [9], the proper choice of the latter is crucial, as already found and explored in Refs. [10–12], for the numerical accuracy, stability and computational efficiency. Generally, two factors determine the accuracy of the SPH method, i.e. smoothing and integration errors [13,1]. While the leading vanishing moments of the kernel function define the smoothing error, the particle summation on neighboring particles

* Corresponding author.

E-mail addresses: zhentong.wang@tum.de (Z. Wang), bo.zhang@tum.de (B. Zhang), oskar.haidn@tum.de (O.J. Haidn), xiangyu.hu@tum.de (X. Hu).<https://doi.org/10.1016/j.jcp.2024.113385>

Received 7 September 2023; Received in revised form 23 August 2024; Accepted 29 August 2024

Available online 2 September 2024

0021-9991/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

within the cut-off radius defines the integration error. In order to achieve higher accuracy of SPH approximation, one can resort to the high-order kernel methods [14,15] using kernels characterized with non-monotonic, non-positive regions and high-order vanishing moments for less smoothing error and sufficiently large compact support for less integration error. However, to avoid the excessive computational effort for large compact supports, the current mainstream kernel functions used in SPH, such as B-spline, Wendland, truncated Gaussian (for compact support) and others, are monotonic and give 2nd-order smoothing error as only the first moment vanishes. Although the truncated Gaussian kernel is a natural choice for SPH, it has yet to be widely used compared to other non-truncated, such as B-spline and Wendland, kernels. One reason is that, to achieve a sufficiently small integration error, the truncated compact support is much larger than that of other non-truncated kernels, resulting in lower computational efficiency. The other reason is that, similarly to the B-spline kernel, it causes the pair-instability problem in which particles tend to appear pair clumping [16,11]. In recent years, the Wendland kernel has gained popularity in the SPH community due to its moderate compact support for sufficient accuracy and ability to overcome the pair-instability problem.

With a given smoothing kernel, besides increasing the size of the compact support, one way to decrease the integration error is using the kernel-correction methods such as higher-order differential operator [17,18], kernel gradient correction [5,19–22], weighted essentially non-oscillatory (WENO) [23–25], moving least squares (MLS) [26–30] and radial basis function (RBF) [31–33]. It has also been found that the integration error strongly relies on particle distribution [13]. Litvinov et al. [1] found that particles relaxed from the random initial position and constant background pressure can achieve the same accuracy as those on uniform lattice positions. Since relaxed particle distribution can be obtained for complex geometries, it is more applicable than uniform lattice distribution for practical applications. Another finding in Ref. [1] is that, under relaxed particle distribution, the convergence rate of integration error (8th-order respected to support radius) can be much higher than that of smoothing error. Therefore, two non-monotonic, namely truncated Laguerre–Gauss and Laguerre–Wendland, kernels with 4th-order smoothing error, have been considered for improving the overall accuracy of SPH approximations. However, these kernels have not been applied for practical SPH algorithms because, for moderate compact support, the actual integration error of the Laguerre–Wendland kernel is much larger than that obtained by the original 2nd-order Wendland kernel. Note that the truncated Laguerre–Gauss kernel was not tested in Ref. [1], probably due to the pair-instability problem of the original truncated Gaussian kernel as mentioned.

In the present work, the large integration error of the Laguerre–Wendland kernel in moderate compact support is revisited, and the source of the error has been analyzed according to the kernel profiles. Based on this, the truncated Laguerre–Gauss kernel is chosen to improve the overall accuracy of SPH approximations. Quite counterintuitively, further numerical tests show that the chosen kernel with moderate compact support avoids the pair-instability problem in the original 2nd-order counterpart and achieves much less relaxation residue than the Wendland kernel. The chosen kernel has been applied to Eulerian and total Lagrangian SPH formulations for fluid and solid dynamics problems. The numerical tests show that considerably higher accuracy has been achieved compared with that of the original Wendland kernel without introducing extra computational effort.

The structure of this paper is as follows: Section 2 gives the preparation of error analysis and revisits the Laguerre–Wendland kernel. Also, the errors of the Laguerre–Wendland and truncated Laguerre–Gauss kernels are analyzed, and the formulation and properties of the latter are discussed in detail. Section 3 introduces standard Eulerian formulation with extensions and total Lagrangian SPH formulation. The extensions include the incorporation of dissipation limiters to decrease numerical dissipation, as well as the utilization of particle relaxation and kernel correction matrix to ensure zero-order and first-order consistency, respectively. Moving on to Section 4, a series of numerical examples are employed to demonstrate the performance and computational efficiency of the proposed kernel. All computational codes utilized in this study have been made publicly available through the SPHinxsys repository [34], accessible at both <https://www.sphinxsys.org> and <https://github.com/Xiangyu-Hu/SPHinxsys>.

2. Kernel analysis

2.1. Preparation of error analysis

The SPH approximation of gradient of a function field $f(\mathbf{r})$ at a particle position $\mathbf{r}_i$ can be derived as following steps

$$\nabla f(\mathbf{r}_i) \approx \int_V \nabla f(\mathbf{r}) W(\mathbf{r}_i - \mathbf{r}, h) dV = - \int_V f(\mathbf{r}) \nabla W(\mathbf{r}_i - \mathbf{r}, h) dV \approx - \sum_j f(\mathbf{r}_j) \nabla W_{ij} V_j, \quad (1)$$

where V_j is the volume of the neighboring particle j respect to particle i , $W(\mathbf{r}_i - \mathbf{r}, h)$ is a smooth kernel function with h denoting the smoothing length and the gradient of the kernel function $\nabla W_{ij} = \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$. The first step is the smoothing approximation, where a smooth kernel function replaces the Dirac delta function and introduces the smoothing error E_s . The second step is integration by parts following the assumption that the kernel function is zero at the domain boundary. The third step is the approximated integration by summation over all neighboring particles, which introduces the integration error E_r . Afterwards, the truncation error E_t is given by $E_t = E_s + E_r$. For simplicity of writing, we simplify $f(\mathbf{r})$ at position $\mathbf{r}_i$ and $\mathbf{r}_j$ to f_i and f_j respectively in the following. As in Ref. [35], Eq. (1) can be modified in a strong form as

$$\nabla f_i = \nabla f_i - f_i \nabla 1 \approx \sum_j f_{ij} \nabla W_{ij} V_j, \quad (2)$$

with $f_{ij} = f_i - f_j$. Also, the modification in a weak form as

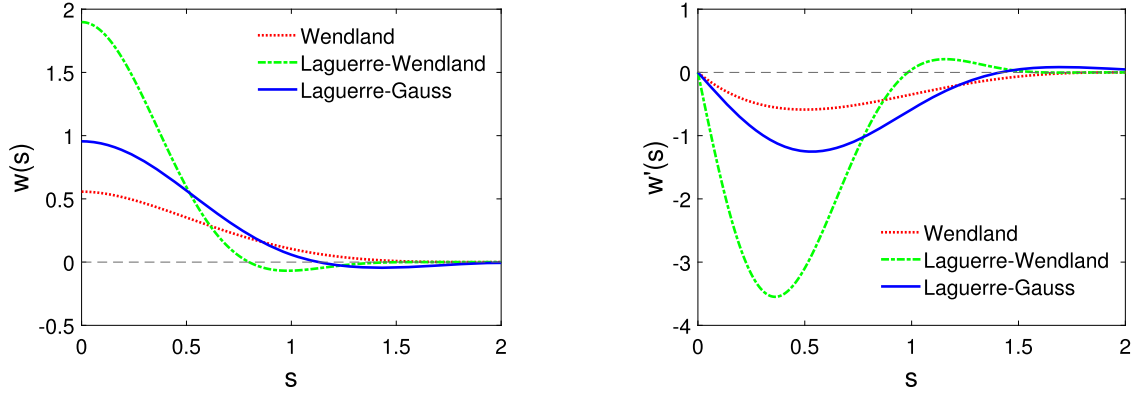


Fig. 1. Laguerre-Gauss kernel and the comparison with the Wendland and Laguerre-Wendland kernels with compact support as $2h$: kernel functions (left panel) and their derivatives (right panel). Here, $C_{LW} \approx 1.898$ within $s \in [0, 2]$ according to the kernel normalization condition. Note that the shallow negative region in the Laguerre-type kernels is for vanishing second-order moment.

$$\nabla f_i = \nabla f_i + f_i \nabla 1 \approx -2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j, \quad (3)$$

with $\bar{f}_{ij} = (f_i + f_j)/2$. Since the weak form is applied in the momentum conservation equations, we employ Eq. (3) to investigate the error estimation in the later section. As is mentioned, the particle distribution plays a crucial role in influencing the integration error [1,36]. Therefore, to achieve a high-quality particle distribution in practical applications and ensure zero-order consistency, i.e. $\sum_j V_j \nabla W_{ij} = \mathbf{0}$, we employ particle relaxation [37] before the error analysis and the acceleration in the process is calculated by

$$\mathbf{a}_i = -\frac{2}{\rho_i} \sum_j V_j \nabla W_{ij}, \quad (4)$$

with ρ denoting the density. Note that the SPH approximation for the derivative of the kernel function directly influences the flux calculation in conservation equations, we further employ the kernel correction matrix [5] given by

$$\mathbf{B}_i = -\left(\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j \right)^{-1} \quad (5)$$

to compensate for the error and satisfy the first-order consistency, which achieves second-order convergence of integration error. Note that there are kernel gradient corrections for even higher-order consistency, such as the mentioned Refs. [19,20,22]. However, they are not employed here due to the high computational cost.

2.2. Error analysis using Laguerre-Wendland kernel

As shown in Litvinov et al. [1], the Laguerre-Wendland and Laguerre-Gauss kernels have the second vanishing moment and 4th-order smoothing error. The former, given by

$$w^{LW}(s) = C_{LW}(1 - \epsilon^2 s^2 + \epsilon^4 s^4/6)(1 - s)_+^6(1 + 6s + 35/3 s^2), \quad (6)$$

where $(\cdot)_+ = \max(0, \cdot)$ and $s = |\mathbf{r}|/h$, a normalization factor $C_{LW} \approx 7.591$, and a scaling factor $\epsilon \approx 2.821$, within $s \in [0, 1]$, had been studied in Litvinov et al. [1] but has not been employed in practice due to its much larger integration error than the widely used Wendland kernel. Here, we revisit this kernel to explore the reason for the excessive integration error. By comparing the profiles of the kernels and their derivatives, as shown in Fig. 1, we find that the gradient magnitude of the Laguerre-Wendland kernel is much larger than that of the Wendland kernel. Note that the SPH approximation highly relies on the size of compact support, which is usually adopted as $2h$ in practice where smoothing length $h = \kappa dp$ with dp denoting the initial particle spacing and κ determining the number of neighboring particles. With these observations, it is straightforward to consider that the possible reason for the Laguerre-Wendland kernel introducing a larger integration error is that the number of neighboring particles is insufficient to resolve the large gradient magnitude within moderate compact support. Based on this, we apply different κ including $\kappa = 1.3, 2.0$ and 2.5 , meaning different total number of neighboring particles for analyzing the integration error. Note that the corresponding resolutions are adopted according to the different κ to keep the smoothing length constant, i.e. the smoothing error is unchanged. For quantitative analysis, following Eq. (3), L_1 -norm and L_{∞} -norm errors are given by

$$\begin{cases} L_1(df/dx) = \frac{1}{N} \sum_i \left| \nabla f_i + 2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j \right| \\ L_{\infty}(df/dx) = \max_i \left| \nabla f_i + 2 \sum_j \bar{f}_{ij} \nabla W_{ij} V_j \right| \end{cases}, \quad (7)$$

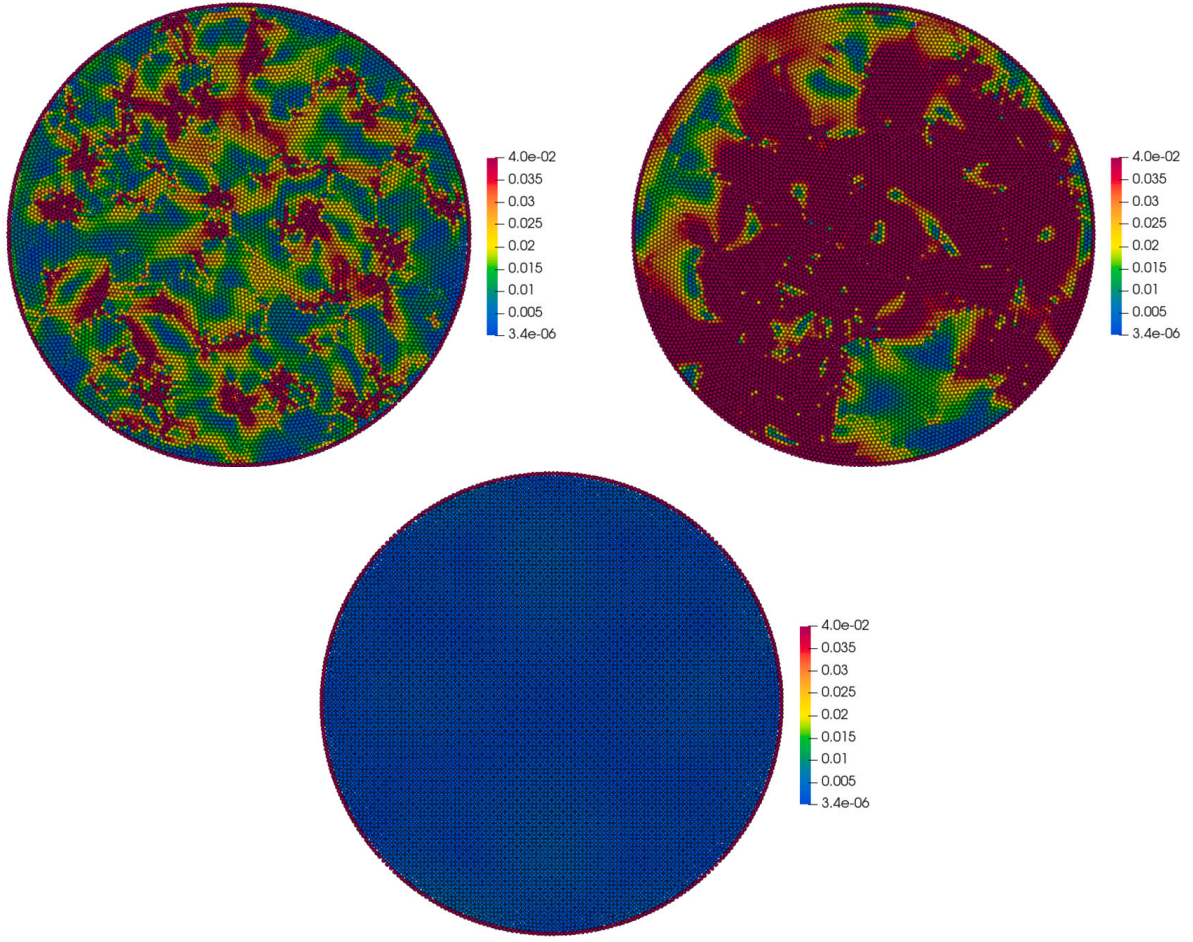


Fig. 2. Particle relaxation residue ranging from 3.4×10^{-6} to 4.0×10^{-2} using the Wendland (top left panel), Laguerre-Wendland (top right panel) and truncated Laguerre-Gauss (bottom panel) kernels at the final relaxation iteration with the spatial resolution $dp = 1/60$. In the case, the boundary particles are not considered due to their not relying on kernel approximation. Based on this, the average relaxation residue and the variation between the maximum and minimum relaxation residues using the truncated Laguerre-Gauss kernel are approximately 0.004 and 0.011, respectively, which are approximately one-tenth and one-twentieth of the values using the Wendland kernel. Meanwhile, the average relaxation residue of 0.07 when employing the Laguerre-Wendland kernel is approximately twenty times larger than that of using the truncated Laguerre-Gauss kernel and the maximum relaxation residue of 1.88 appears to be unreasonable. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 1

Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the Wendland and Laguerre-Wendland kernels with different κ without the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Wendland	1.3	0.034	0.123	0.259	0.400
Laguerre-Wendland		0.070	0.673	1.880	3.506
Wendland	2.0	0.014	0.073	0.027	0.150
Laguerre-Wendland		0.010	0.018	0.028	0.046
Wendland	2.5	0.018	0.078	0.032	0.161
Laguerre-Wendland		0.005	0.009	0.017	0.030

with N and df/dx denoting the total number of particles and the derivative of the function field, respectively. Here, the errors obtained in Eq. (7) are the truncation errors.

The study applies a circular geometry of a diameter $D = 2$ with the resolution $dp = 1/60$. We test the SPH approximation for the derivatives of a constant function $f = 1$ and a trigonometric function $f = \sin(2\pi x)$ using the Wendland and Laguerre-Wendland kernels without and with the kernel correction matrix in this domain under the respective particle distributions obtained by their respective kernels shown in Fig. 2, and Tables 1 and 2 list the truncation errors with different κ . Note that the boundary particles are not considered in the kernel analysis because these particles are given the assigned values and do not evaluate the kernel gradient. From the results, the truncation errors using the Laguerre-Wendland kernel exhibit similar behavior to those obtained with the

Table 2

Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the Wendland and Laguerre-Wendland kernels with different κ with the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Wendland	1.3	0.039	0.052	0.209	0.187
Laguerre-Wendland		0.077	0.067	2.334	2.319
Wendland	2.0	0.014	0.042	0.026	0.067
Laguerre-Wendland		0.011	0.007	0.029	0.028
Wendland	2.5	0.018	0.043	0.033	0.068
Laguerre-Wendland		0.005	0.003	0.017	0.015

Wendland kernel, declining with $\kappa \leq 2$ and remaining relatively unchanged at $\kappa = 2.5$. These results suggest that, while the smoothing error dominates at $\kappa \geq 2$, the integration error takes the lead at $\kappa = 1.3$. For the condition of smoothing error dominant with $\kappa \geq 2$, it is implied that the truncation error of the Laguerre-Wendland kernel is less than the Wendland kernel, as shown in Table 2, which agrees with the property of kernels' smoothing order. In addition, given the reasonable truncation errors, while the impact of the kernel correction matrix on reducing the truncation error with the derivative of $f = 1$ is not as apparent, this matrix can further minimize the error with that of $f = \sin(2\pi x)$ effectively, suggesting that the kernel correction matrix has distinct effects on higher-order terms.

2.3. Truncated Laguerre-Gauss kernel

Another 4th-order kernel, as shown in Fig. 1, the Laguerre-Gauss kernel, with a smaller gradient magnitude than the Laguerre-Wendland kernel, is considered a potential option for not introducing excessive integration errors.

2.3.1. Kernel profile

The expression of the Laguerre-Gauss kernel [1] is given by

$$w(s) = \pi^{-1} L_1^{(1)}(s^2) e^{-s^2}, \quad (8)$$

where $L_1^{(1)}$ denotes the generalized Laguerre polynomial with a non-negative Fourier transform and vanishing second-order moment. For implementing the kernel in practice, we truncate the compact domain to $2h$ and give the expression for the truncated Laguerre-Gauss kernel as

$$w(s) = \alpha_d \begin{cases} (1 - \frac{s^2}{2} + \frac{s^4}{6}) e^{-s^2} & \text{if } 0 \leq s \leq 2 \\ 0 & \text{if } s > 2 \end{cases}, \quad (9)$$

where α_d is the normalized coefficient in d dimensional space with the value of $\alpha_1 = 8/(5\sqrt{\pi})$, $\alpha_2 = 3/\pi$ and $\alpha_3 = 8/\pi^{3/2}$. Also, the truncated Laguerre-Gauss kernel and its derivative and the comparisons with the Wendland, as well as the Laguerre-Wendland kernels, are presented in Fig. 1, which shows that its gradient magnitude is larger than that of the Wendland kernel but smaller than that of the Laguerre-Wendland kernel and the value of kernel function at $2h$ is close to 0. In numerical simulations, to avoid adding extra computational effort, we apply the compact support as $2h$ with smoothing length $h = 1.3dp$, meaning that the computational efficiency using the truncated Laguerre-Gauss kernel is the same as that using the Wendland kernel.

2.3.2. Performance in stability

To exploit the properties of the truncated Laguerre-Gauss kernel, we first investigate and compare the relaxed particle distribution using the Wendland, Laguerre-Wendland and truncated Laguerre-Gauss kernels in the same circular domain mentioned above. Fig. 2 shows the relaxation residue ranging from 3.4×10^{-6} to 4.0×10^{-2} using the Wendland, Laguerre-Wendland and truncated Laguerre-Gauss kernels at the final relaxation iteration. Utterly contrary to our intuition that the truncated Laguerre-Gauss kernel is similar to the original truncated Gaussian kernel in that it suffers from the pair-instability problem with $\kappa = 1.3$, it can be observed that the particle relaxation using the truncated Laguerre-Gauss kernel does not exhibit this problem. Similarly with the kernel analysis in section 2.2, the particles in the boundary are also omitted in the following analysis. Based on this, the truncated Laguerre-Gauss kernel yields an average relaxation residue of 0.004, and a variation between the maximum and minimum relaxation residues of 0.011 and these values represent roughly one-tenth and one-twentieth of those obtained using the Wendland kernel. Besides, the Laguerre-Wendland kernel results in an average relaxation residue of 0.07, approximately twenty times larger than that observed with the truncated Laguerre-Gauss kernel. However, the maximum relaxation residue of 1.880 using the Laguerre-Wendland kernel is unacceptable and unreasonable. Therefore, the truncated Laguerre-Gauss kernel ensures numerical stability and obtains higher-quality particle distribution. As widely acknowledged, utilizing high-order kernels with negative regions can lead to stability issues [1], particularly for updated Lagrangian SPH methods. Nevertheless, employing these kernels in the Eulerian or total Lagrangian SPH methods mitigates this issue, as demonstrated in Fig. 2, owing to the fixed and high-quality particle configuration. Further validation of its stability will be carried out through a series of fluid and solid cases in Section 4.

Table 3

Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the truncated Laguerre-Gauss kernel with different κ without the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Laguerre-Gauss	1.1	0.021	0.259	0.614	1.253
	1.3	0.004	0.123	0.011	0.204
	1.6	0.007	0.210	0.822	0.817

Table 4

Truncation errors of the SPH approximation for df/dx with $f = 1$ and $f = \sin(2\pi x)$ using the truncated Laguerre-Gauss kernel with different κ with the kernel correction matrix.

Truncation errors	κ	$L_1(df/dx)$		$L_{\infty}(df/dx)$	
		$f = 1$	$f = \sin(2\pi x)$	$f = 1$	$f = \sin(2\pi x)$
Laguerre-Gauss	1.1	0.021	0.061	0.603	0.448
	1.3	0.004	0.004	0.011	0.009
	1.6	0.006	0.024	0.784	0.364

2.3.3. Performance in accuracy

Similarly, with the analysis of the Laguerre-Wendland kernel, we test the SPH approximation for the derivatives of the same functions using the truncated Laguerre-Gauss kernel to further verify the approximation accuracy. Specifically, the truncation error using the derivative of function $f = 1$ equals the relaxation residue in Eq. (4).

Tables 3 and 4 list the truncation errors using the truncated Laguerre-Gauss kernel. From the results, the truncation error in the case of the insufficient number of neighboring particles with $\kappa = 1.1$ is still relatively large. Besides, the large errors obtained with $\kappa = 1.6$ are due to the pair instability when the number of neighboring particles is too large. Note that $\kappa = 1.3$ is applied in practice to avoid extra computational effort. In the case of $\kappa = 1.3$ with the derivative of the trigonometric function, we observe that the truncation errors using the truncated Laguerre-Gauss kernel in Table 3 and Wendland kernel in Table 1 are comparable without the utilization of the kernel correction matrix. However, the truncation error using the truncated Laguerre-Gauss kernel in Table 4 is significantly smaller than the errors using the Wendland and Laguerre-Wendland kernels in Table 2, demonstrating that the former kernel's cooperation with the kernel correction matrix offers much higher accuracy. This finding suggests the integration error using the truncated Laguerre-Gauss kernel is smaller than the other two kernels as the integration errors are dominant in using the Wendland and Laguerre-Wendland kernels with $\kappa = 1.3$. Furthermore, the proposed kernel exhibits superior performance with respect to the average relaxation residue and the variation of the maximum and minimum residue. Specifically, in Tables 3, the average relaxation residue and the variation are 0.004 and 0.011, respectively, which are approximately one-tenth and one-twentieth of the values listed in Tables 1 for the Wendland kernel, proving that the truncated Laguerre-Gauss kernel enables obtaining much less average relaxation residue and the variation than those using the other two kernels and these results agree with the particle distribution shown in Fig. 2. Note that we choose the compact support of $2h$ for the Laguerre-Gauss kernel, on the one hand, to match that of the Wendland kernel for the same computation efficiency. On the other hand, such a choice is nearly optimal because our primitive numerical experiments (not shown here) suggest generally inferior results if larger compact supports are used. This may be because the relaxed particle distribution with large compact supports resembles that obtained by the Wendland kernel, as shown in Fig. 2, other than the lattice-like one.

2.3.4. Convergence analysis

In this section, we employ the same circular geometry mentioned earlier to conduct a convergence study, initializing a scalar field according to the function expressed as

$$f(x) = e^{-x^2/0.1}. \quad (10)$$

Following Refs. [23,38], the L_2 -norm error of the approximated gradient operators can be defined as

$$L_2(df/dx) = \frac{\sqrt{\sum_i |(\nabla f_i)^{Theoretical} - (\nabla f_i)^{Numerical}|^2}}{\sqrt{\sum_i |(\nabla f_i)^{Theoretical}|^2}}, \quad (11)$$

where $(\nabla f_i)^{Numerical}$ is the strong form as Eq. (2).

Fig. 3 illustrates the convergence study utilizing the Wendland and truncated Laguerre-Gauss kernels across various particle resolutions with $\kappa = 1.3$ unchanged. It is observed that, without the kernel correction matrix, as resolution increases, the convergence rate of the Wendland kernel begins at second-order and gradually degenerates to saturation, where the integration error dominates.

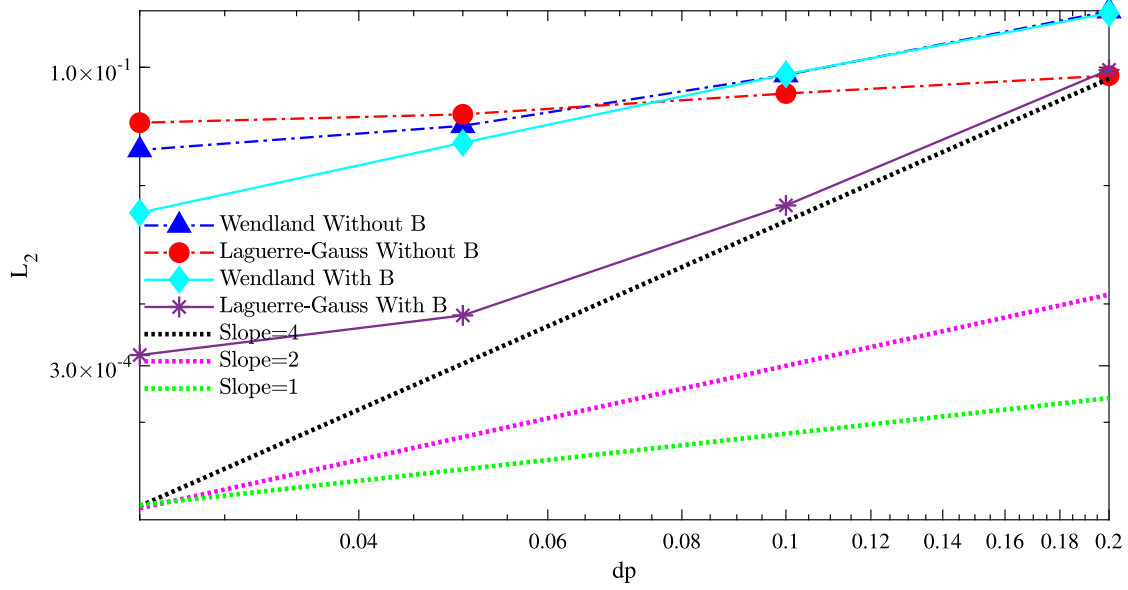


Fig. 3. Convergence study using the Wendland and Laguerre-Gauss kernel with different resolutions by comparing the different theoretical order curves.

It is also observed that while the truncated Laguerre-Gauss kernel gives considerably higher accuracy than the Wendland kernel at low resolution, its accuracy degenerates to saturation quickly with increasing resolution. Such behavior indicates that the truncated Laguerre-Gauss kernel obtains higher accuracy at low resolution due to the smaller smoothing error, and its integration error is still larger than that of the Wendland kernel.

With the kernel correction matrix, it is observed that while the Wendland kernel sustains second-order convergence, the truncated Laguerre-Gauss kernel exhibits fourth-order convergence at low resolution but degenerates to second-order with increasing resolutions. Such behaviors are expected since both the smoothing and integration errors of the Wendland kernel, in this case, convergence at second order, and, for the truncated Laguerre-Gauss kernel, the fourth-order smoothing error dominates at low resolution and the second-order integration error at high-resolution.

The significance of the convergence study is that, as shown in Fig. 3, independent of particle resolution, with the kernel correction matrix, the truncated Laguerre-Gauss kernel always gives much smaller errors than the Wendland kernel. Note that such improvement is achieved with the same support and kernel correction matrix as the Wendland kernel. Furthermore, at high resolution, the truncated Laguerre-Gauss kernel requires only a small fraction of particles (about 1/16 in two dimensions, as shown in Fig. 3) to achieve the same accuracy as the Wendland kernel does. Also note that our objective here is to improve numerical accuracy simply by using a different kernel without increasing computational cost, other than the approaches aiming at sustained high-order (higher than second-order) convergence rate, which may achieve even higher accuracy but require much more elaborated efforts and computational cost.

3. Governing equations and SPH methods

Here, we consider an extended Eulerian SPH method for fluid dynamics [3] and the standard total Lagrangian SPH method for solid mechanics [4,39].

3.1. Extended Eulerian SPH

3.1.1. Governing equations

The conservation equations in the Eulerian framework can be described by

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (12)$$

where $\mathbf{U}$ is the vector of the conserved variables, $\mathbf{F}(\mathbf{U})$ the corresponding fluxes. Here, they can be expressed specifically as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{bmatrix} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(E + p) \end{bmatrix} + \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(E + p) \end{bmatrix}, \quad (13)$$

where u , v and w are the components of velocity, ρ and p denote the density and pressure, respectively. Here, $E = \frac{\rho v^2}{2} + \rho e$ is the total energy per volume with e the internal energy. The equation of state (EOS) is added to close the Eq. (12) by

$$p = \begin{cases} \rho(\gamma - 1)e & \text{For compressible flows} \\ c^2(\rho - \rho_0) & \text{For weakly-compressible flows} \end{cases}, \quad (14)$$

where γ is the heat capacity ratio, ρ_0 the reference density. Following the ideal gas equation in compressible flows and the weakly-compressible assumption in incompressible flows, the speed of sound is derived by

$$c = \begin{cases} \sqrt{\frac{\gamma p}{\rho}} & \text{For compressible flows} \\ 10U_{max} & \text{For weakly-compressible flows} \end{cases}. \quad (15)$$

Here, U_{max} the maximum velocity in the flow field to control the density variation less than 1%. Note that the energy equation is turned off in weakly-compressible flows. In the viscous weakly-compressible flows, we add a viscous force given by

$$\left(\frac{\partial \mathbf{v}}{\partial t}\right)^{(v)} = \eta \nabla^2 \mathbf{v} \quad (16)$$

to the momentum equation.

3.1.2. Standard Eulerian SPH discretization

Following Ref. [40], the discretization form of the Eq. (12) can be written as

$$\begin{cases} \frac{\partial}{\partial t} (V_i \rho_i) + 2V_i \sum_j V_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} = 0 \\ \frac{\partial}{\partial t} (V_i \rho_i \mathbf{v}_i) + 2V_i \sum_j V_j \left[(\rho \mathbf{v} \otimes \mathbf{v})_{E,ij}^* + p_{E,ij}^* \mathbb{I} \right] \cdot \nabla W_{ij} = 0, \\ \frac{\partial}{\partial t} (V_i E_i) + 2V_i \sum_j V_j \left[(E \mathbf{v})_{E,ij}^* + (p \mathbf{v})_{E,ij}^* \right] \cdot \nabla W_{ij} = 0 \end{cases} \quad (17)$$

where $\mathbf{v}$ represents the velocity, $\mathbb{I}$ the identity matrix and terms $(\cdot)_{E,ij}^*$ are solution of the Riemann problem [40]. Here, $\nabla W_{ij} = \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}$ denotes kernel gradient where $\mathbf{e}_{ij} = -\mathbf{r}_{ij}/r_{ij}$ with $\mathbf{r}_{ij}$ the displacement pointing from particle j to i .

To obtain the solution to the Riemann problem, three waves with the smallest speed S_l , middle speed S_* and largest speed S_r are utilized. Note that the middle wave distinguishes the two intermediate states as (ρ_l^*, u_l^*, p_l^*) and (ρ_r^*, u_r^*, p_r^*) . For compressible flows, we employ the HLLC Riemann solver [41,42] because of its ability to capture the shock discontinuity, with the three wave speeds estimated as

$$\begin{cases} S_l = u_l - c_l \\ S_r = u_r + c_r \\ S_* = \frac{\rho_r u_r (S_r - u_r) + \rho_l u_l (u_l - S_l) + p_l - p_r}{\rho_r (S_r - u_r) + \rho_l (u_l - S_l)} \end{cases}. \quad (18)$$

Then, the intermediate states can be calculated as

$$\begin{cases} p^* = p_l + \rho_l (u_l - S_l) (u_l - u^*) = p_r + \rho_r (S_r - u_r) (u^* - u_r) \\ \mathbf{v}^* = u^* \mathbf{e}_{ij} + \left[\frac{1}{2} (\mathbf{v}_l + \mathbf{v}_r) - \frac{1}{2} (u_l + u_r) \mathbf{e}_{ij} \right] \\ \rho_{l/r}^* = \rho_{l/r} \frac{(S_{l/r} - q_{l/r})}{(S_{l/r} - u^*)} \\ E_{l/r}^* = \frac{(S_{l/r} - q_{l/r}) E_{l/r} - p_{l/r} q_{l/r} + p^* u^*}{S_{l/r} - u^*} \end{cases}, \quad (19)$$

where q represents the velocity magnitude and $u^* = S_*$. In weakly-compressible flows, the intermediate states follow the assumption satisfying $p_l^* = p_r^* = p^*$ and $u_l^* = u_r^* = u^*$ and then a linearised Riemann solver can be expressed as [43,25]

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (20)$$

with $\bar{\rho}$ and $\bar{c}$ interface-particle averages.

3.1.3. Eulerian SPH extensions

In this section, we implement the particle relaxation following Eq. (4) and kernel correction matrix following Eq. (5) and detail the dissipation limiters, which are used to improve the stability and accuracy of the Eulerian SPH method. To maintain the conservation property, the corrected kernel gradient can be re-evaluated as

$$\nabla' W_{ij} = \frac{\mathbf{B}_i + \mathbf{B}_j}{2} \nabla W_{ij} \quad (21)$$

to replace the original kernel gradient ∇W_{ij} in Eq. (17).

In order to reduce the numerical dissipation introduced by the Riemann problem, dissipation limiters [25,44] are implemented and can significantly improve numerical accuracy. In the HLLC Riemann solver, the middle wave speed and pressure in Eq. (19) with the limiter can be rewritten as

$$\begin{cases} u^* = \frac{\rho_l u_l c_l + \rho_r u_r c_r}{\rho_l c_l + \rho_r c_r} + \frac{p_l - p_r}{\rho_l c_l + \rho_r c_r} \beta_{HLLC}^2 \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{HLLC} [\rho_r c_r (u^* - u_r) - \rho_l c_l (u_l - u^*)] \end{cases}, \quad (22)$$

with the limiter as

$$\beta_{HLLC} = \min \left(\eta_{HLLC} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right), \quad (23)$$

where $\eta_{HLLC} = 1.0$ according to the numerical tests. Also, the linearised Riemann solver in Eq. (20) with the limiter can be rewritten as

$$\begin{cases} u^* = \frac{u_l + u_r}{2} + \frac{1}{2} \beta_{linearisd}^2 \frac{(p_l - p_r)}{\bar{\rho} \bar{c}} \\ p^* = \frac{p_l + p_r}{2} + \frac{1}{2} \beta_{linearisd} \bar{\rho} \bar{c} (u_l - u_r) \end{cases}, \quad (24)$$

with the dissipation limiter $\beta_{linearisd}$ as

$$\beta_{linearisd} = \min \left(\eta_{linearisd} \max \left(\frac{u_l - u_r}{\bar{c}}, 0 \right), 1 \right). \quad (25)$$

where $\eta_{linearisd} = 15.0$.

3.2. Total Lagrangian SPH

In total Lagrangian SPH framework for elastic solid dynamics, the mass and momentum conservation equations can be expressed as

$$\begin{cases} \rho = J^{-1} \rho^0 \\ \rho^0 \frac{d\mathbf{v}_i}{dt} = \nabla^0 \cdot \mathbb{P}^T \end{cases}, \quad (26)$$

with ρ^0 and ρ the initial and current density, respectively. Here, $J = \det(\mathbb{F})$ with $\mathbb{F}$ denoting the deformation gradient tensor, $\mathbf{v}$ represents the solid velocity, and $\mathbb{P}$ as well as T are the first Piola-Kirchhoff stress tensor and the matrix transposition operator, respectively. Also, $\mathbb{P}$ can be derived as $\mathbb{P} = \mathbb{F}\mathbb{S}$ with $\mathbb{S}$ the second Piola-Kirchhoff stress tensor where the constitutive equation [34] when the material is linearly elastic and isotropic is given by

$$\mathbb{S} = \lambda \text{tr}(\mathbb{E})\mathbb{I} + 2G\mathbb{E}, \quad (27)$$

with λ and G denoting the Lamé parameter and the shear modulus, respectively.

Then, the momentum equation in Eq. (26) with the weak form [34] of the SPH particle approximation can be discretized as

$$\rho_i^0 \frac{d\mathbf{v}_i}{dt} = \sum_j \left(\mathbb{P}_i \mathbf{B}_i^{0T} + \mathbb{P}_j \mathbf{B}_j^{0T} \right) \cdot \nabla^0 W_{ij} V_j^0, \quad (28)$$

where $\nabla^0 W_{ij} = \frac{\partial W(r_{ij}^0, h)}{\partial r_{ij}^0} \mathbf{e}_{ij}^0$ and $\mathbf{B}^0$ are the kernel gradient and correction matrix in the initial configuration, respectively. In addition, the deformation change rate is updated with the strong form of the SPH particle approximation as follows:

$$\frac{d\mathbb{F}}{dt} = \sum_j V_j^0 (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla^0 W_{ij} \mathbf{B}_i^0. \quad (29)$$

4. Numerical results

In this section, a set of numerical simulations, including fluid dynamics and elastic solid dynamics, are tested to verify the accuracy and computational efficiency of the truncated Laguerre-Gauss kernel (denoted as Laguerre-Gauss) and comparison with the Wendland kernel [45] (denoted as Wendland). For clarity, the cut-off radius of both kernels is $2.6dp$.

4.1. The Shu-Osher problem

In this part, a one-dimensional Shu-Osher problem is studied to investigate the accuracy of the Laguerre-Gauss kernel in the compressible flow. The initial condition is given by

$$(\rho, u, v, p) = \begin{cases} (\frac{27}{7}, \frac{4\sqrt{35}}{9}, 0, \frac{31}{3}) & x \leq 1 \\ (1 + 0.2 \sin 5x, 0, 0, 1) & \text{otherwise} \end{cases}, \quad (30)$$

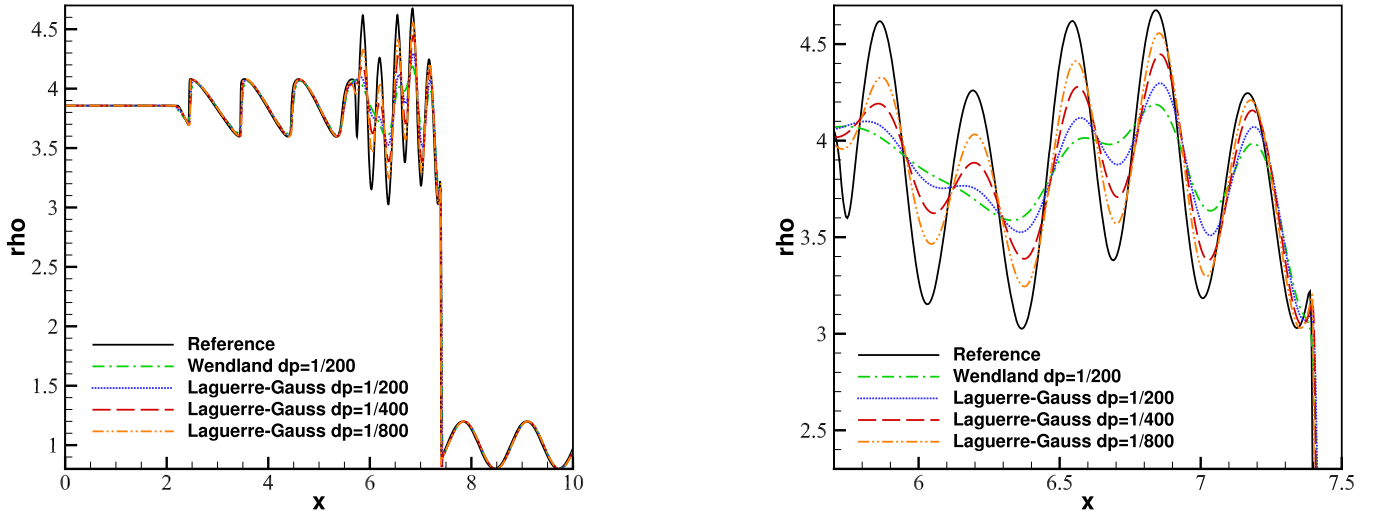


Fig. 4. The Shu-Osher problem: Density profile (left panel) and zoom-in view (right panel) using the Wendland and Laguerre-Gauss kernels and their comparisons against the reference data [46].

with the domain $x \in [0, 10]$ and final time $t = 1.8$. Also, to facilitate a convergence analysis, three spatial resolutions $dp = 1/200$, $1/400$ and $1/800$ are applied across each segment of the domain length.

Fig. 4 shows the density and zoom-in view profiles using the Wendland and Laguerre-Gauss kernels and the comparisons with the reference data [46]. It can be observed that the Laguerre-Gauss kernel can significantly improve the accuracy compared to the Wendland kernel at the resolution $dp = 1/200$. Also, the results converge rapidly as the resolution increases, and there is roughly first-order convergence at the extremities and second-order convergence in regions other than the extremities. Note that, even though there is a non-positive region in the profile of the Laguerre-Gauss kernel, the solution is quite stable, and no noticeable oscillations or the Runge or Gibbs phenomena [47] observed. This may be due to the numerical dissipation introduced by the Riemann solvers. Such numerical stability is further validated in the next section by the classic double Mach reflection problem.

4.2. Double Mach reflection of a strong shock

In this section, we test a two-dimensional phenomenon, the double Mach reflection of a strong shock. Following Ref. [48], we define the computational domain as $(x, y) \in [0, 4] \times [0, 1]$. The initial condition is specified as follows:

$$(\rho, u, v, p) = \begin{cases} (1.4, 0, 0, 1) & x \geq \frac{1}{6} + \frac{y}{\sqrt{3}}, \\ (8, 8.25 \cos(\frac{\pi}{6}), -8.25 \sin(\frac{\pi}{6}), 116.5) & \text{otherwise} \end{cases}, \quad (31)$$

and the simulation is executed until the final time $t = 0.2$. A right-moving Mach 10 shock is situated initially at $(x, y) = (\frac{1}{6}, 0)$, forming a 60-degree angle with the x -axis. The bottom boundary is a reflective wall boundary, extending from $x = 1/6$ to $x = 4$. The left-hand boundary is characterized by the post-shock boundary condition, while a zero-gradient condition is enforced on the right boundary at $x = 4$. For this simulation, the spatial resolution is set to $dp = 1/256$ to discretize the computational domain.

Fig. 5 displays the density contours, ranging from 0.9 to 24.0, utilizing the Wendland (top panel) and Laguerre-Gauss (bottom panel) kernels. It is observed that, again, both kernels yield stable and smooth density profiles, suggesting quite good numerical stability, and the Laguerre-Gauss kernel obtains significantly more accurate results characterized by the sharper shock fronts and shear layer.

4.3. Taylor-Green vortex flow

This section considers the two-dimensional viscous Taylor-Green vortex flow to validate the proposed kernel in the weakly-compressible flow. Following Ref. [49], the initial velocity in a unit domain with the periodic boundary conditions in x - and y -directions is given by

$$\begin{cases} u(x, y, t) = -\exp^{bt} \cos(2\pi x) \sin(2\pi y), \\ v(x, y, t) = \exp^{bt} \sin(2\pi x) \cos(2\pi y), \end{cases}, \quad (32)$$

where the decay rate is $b = -8\pi^2/Re$, the Reynolds number $Re = 100$, the total kinetic energy decay rate $-16\pi^2/Re$ and the final time $t = 5$. Also, we apply three resolutions, including $dp = 1/25$, $1/50$ and $1/100$, to test the convergence study.

Fig. 6 portrays the kinetic energy decay using the Wendland and Laguerre-Gauss kernels and the comparison with the theoretical solution (denoted as “Exact”). From the curves of kinetic energy decay, the method using the Laguerre-Gauss kernel shows improved accuracy by reducing the numerical dissipation more effectively at a resolution of $dp = 1/25$, and the results achieve second-order convergence as the resolution increases.

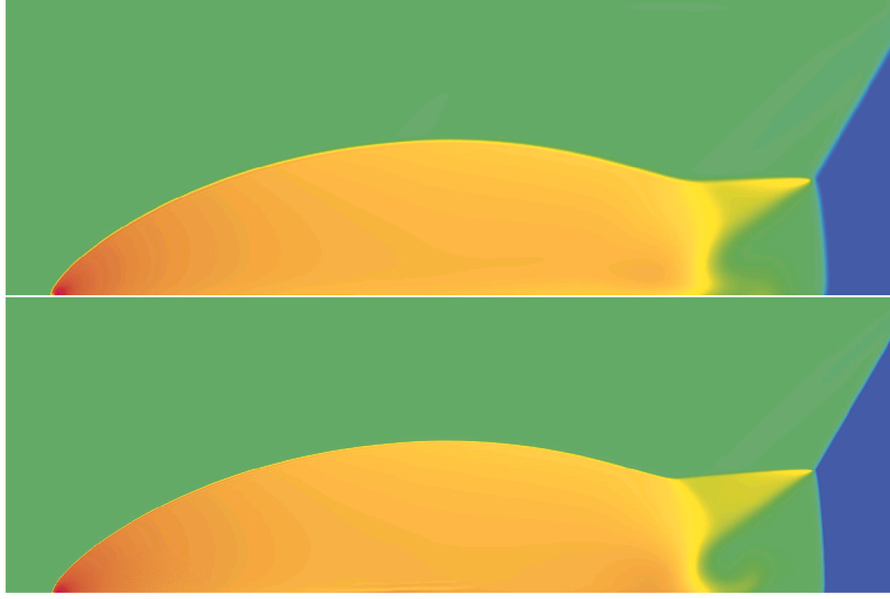


Fig. 5. Double Mach reflection of a strong shock: Density contour ranging from 0.9 to 24.0 obtained by the Wendland (top panel) and Laguerre-Gauss (bottom panel) kernels with the resolution $dp = 1/256$ at the final time $t = 0.2$.

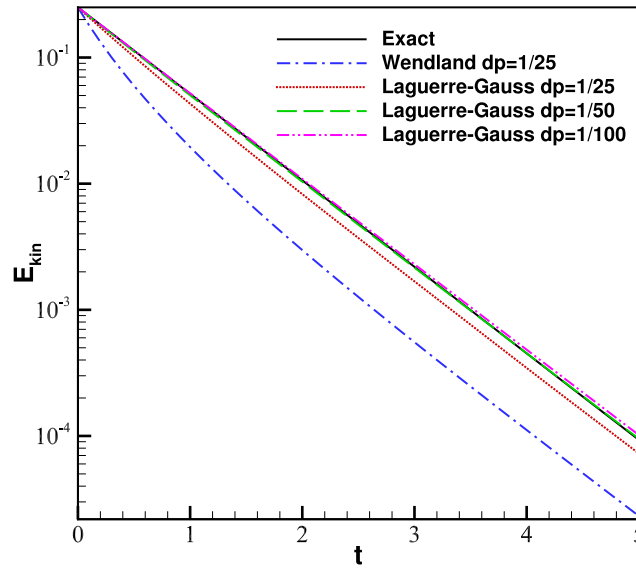


Fig. 6. Taylor-Green vortex flow ($Re = 100$): Decay of the kinetic energy using the Wendland and Laguerre-Gauss kernels and the comparisons with the theoretical solution (denoted as “Exact”).

4.4. Flow around a cylinder

In this section, flow around a cylinder case containing complex geometries is studied to investigate the versatility of the proposed kernel. In order to assess the numerical results quantitatively, the drag and lift coefficients are given as

$$C_D = \frac{2F_D}{\rho_\infty u_\infty^2 A}, C_L = \frac{2F_L}{\rho_\infty u_\infty^2 A}, \quad (33)$$

where F_D and F_L denote the drag and lift force on the cylinder, respectively, and velocity u_∞ as well as density ρ_∞ in the far-field are set as 1. Also, the Strouhal number $St = fD/u_\infty$ in the unsteady cases and the Reynolds numbers $Re = \rho_\infty u_\infty D/\mu$ is 100 with the cylinder diameter $D = 2$ in the case where the far-field boundary conditions are applied in all boundaries and the computational time is $t = 300$. To alleviate the effects of far-field boundary conditions on the simulations, the large computational domain size is set as $[25D, 15D]$, where the location of the cylinder center is $(7.5D, 7.5D)$. The final time is $t = 300$, and the spatial resolutions are applied as $dp = 1/10, 1/20$, and $1/30$ for the convergence study.

Fig. 7 shows the velocity contour ranging from 1.8×10^{-3} to 1.5 using the Wendland and Laguerre-Gauss kernels with the Reynolds number $Re = 100$ under the resolution $dp = 1/6$ at $t = 300$. It can be seen that the method using the Laguerre-Gauss kernel efficiently

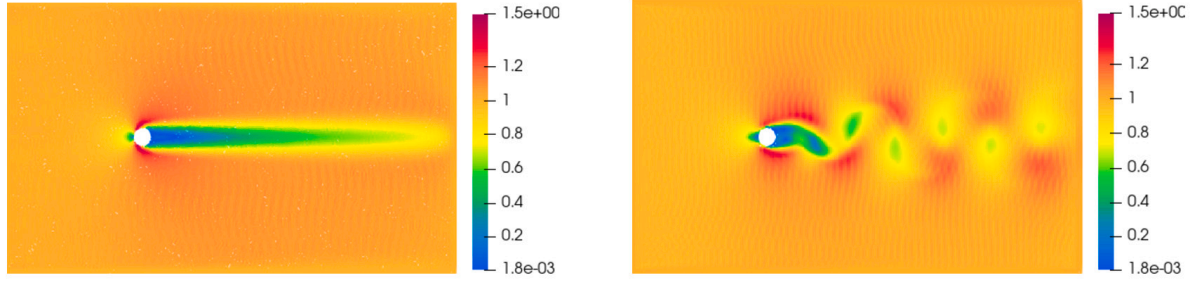


Fig. 7. Flow around a cylinder: Velocity contour ranging from 1.8×10^{-3} to 1.5 using the Wendland (left panel) and Laguerre-Gauss (right panel) kernels with $Re = 100$ under the resolution $dp = 1/6$ at $t = 300$.

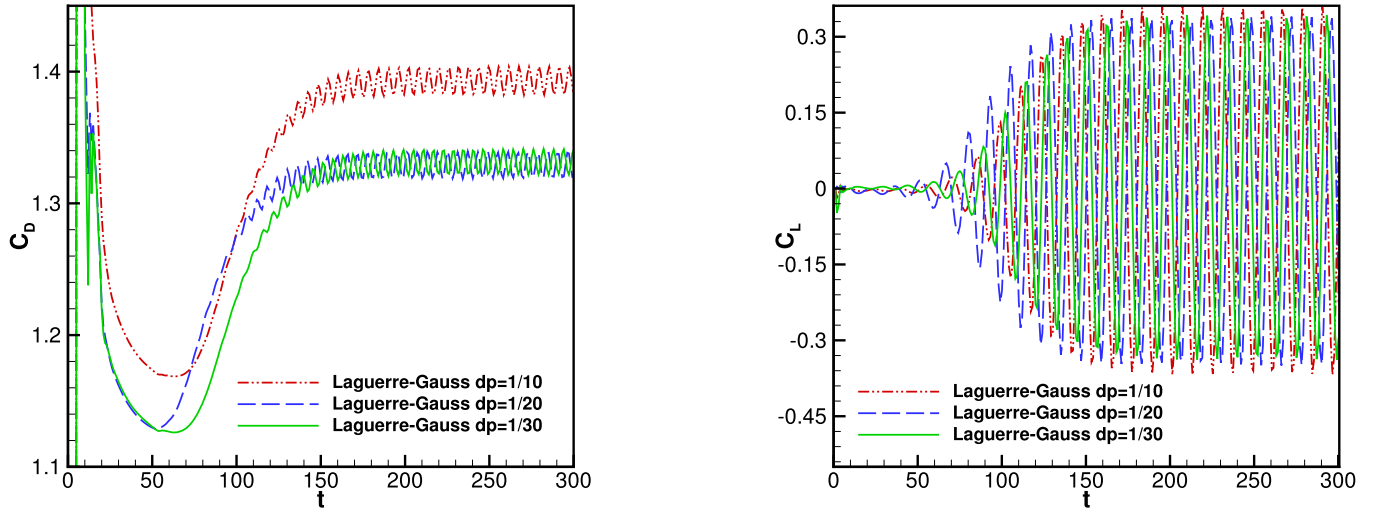


Fig. 8. Flow around a cylinder: Drag (left panel) and lift (right panel) coefficients with the time using the Laguerre-Gauss kernel with $Re = 100$.

Table 5

Flow around a cylinder: Drag and lift coefficients from different experimental and simulation results at $Re = 100$ and the result using the Laguerre-Gauss kernel with the resolution $dp = 1/30$.

Parameters	C_D	C_L	S_l
White [50]	1.46	-	-
Khademinejad et al. [51]	$1.30 \pm -$	± 0.292	0.152
Chiu et al. [52]	1.35 ± 0.012	± 0.303	0.166
Le et al. [53]	1.37 ± 0.009	± 0.323	0.160
Brehm et al. [54]	1.32 ± 0.010	± 0.320	0.165
Liu et al. [55]	1.35 ± 0.012	± 0.339	0.165
Present	1.33 ± 0.009	± 0.340	0.169

captures the vortex street while that using the Wendland kernel fails to present the same physical phenomenon, indicating that the proposed kernel is more effective in reducing numerical errors and thus obtaining more accurate results.

We further verify the correctness of the proposed kernel with three resolutions by comparing convergent drag and lift coefficients with other experimental and simulation results listed in Table 5 where the result of using extended Eulerian SPH with the Laguerre-Gauss kernel is denoted as “Present”. Fig. 8 presents the drag and lift coefficients calculated by the Eulerian SPH method using the Laguerre-Gauss kernel under three resolutions from $dp = 1/10$ to $dp = 1/30$ until the final time $t = 300$, showing that the drag coefficients reach a stable value after a period of fluctuation at the beginning while the lift coefficient oscillates around zero. From Fig. 8 and Table 5, it can be obtained that the drag and lift coefficients have converged and are in agreement with other references, indicating that the results can be considered correct. In the present study, the computations are all performed on an Intel Core i7-10700 2.90 GHz 8-core desktop computer and the CPU wall-clock times obtained by extended Eulerian SPH method using the Wendland and Laguerre-Gauss kernels with the spatial resolution $dp = 1/10$ are 4604.81 s and 4818.91 s in the whole process, respectively, indicating that the computational efficiency of using both kernels is comparable.

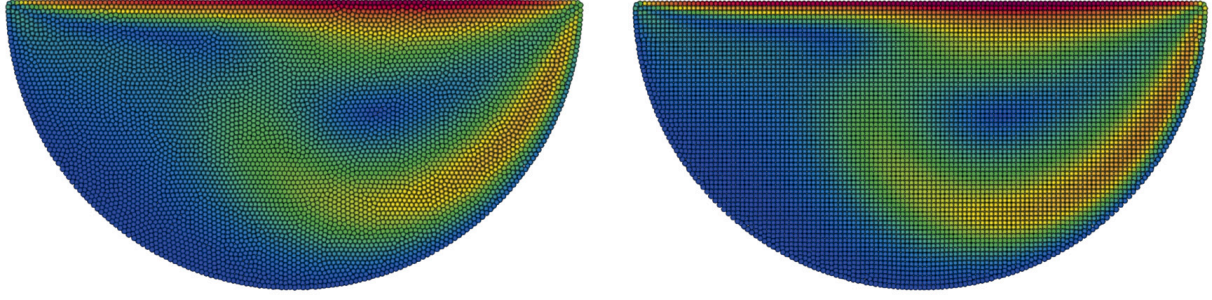


Fig. 9. Lid-driven semi-circular cavity flow with $Re = 1000$: The velocity magnitude contours ranging from 0 to 0.97 obtained with the Wendland (left panel) and Laguerre-Gauss (right panel) kernels with the resolution $dp = 1/129$.

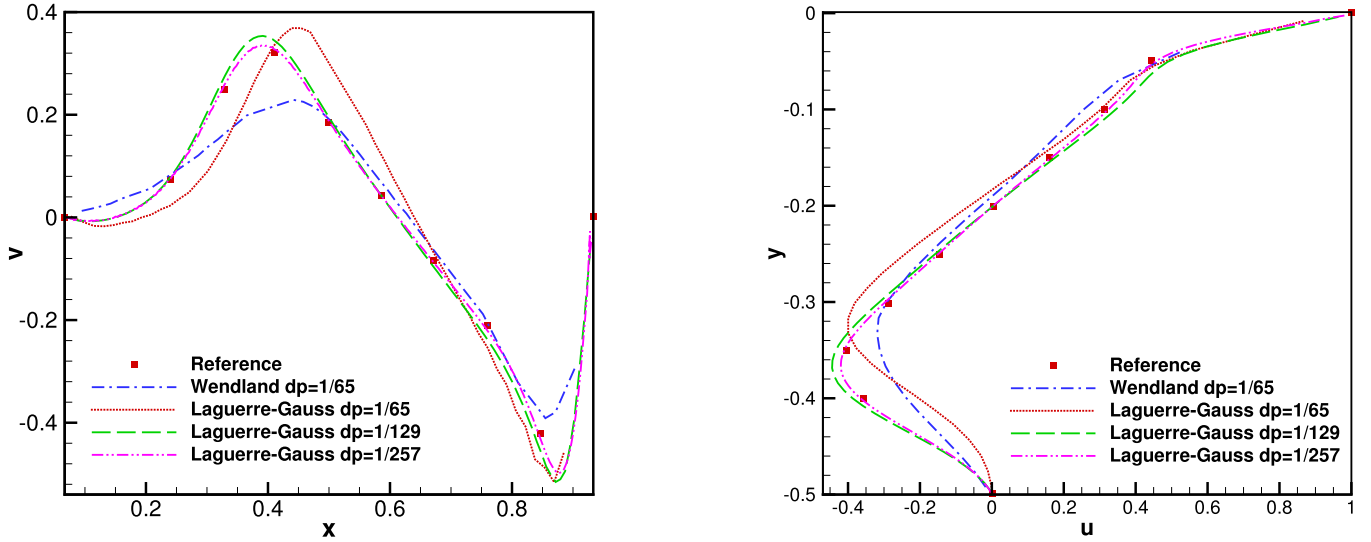


Fig. 10. Lid-driven semi-circular cavity flow with $Re = 1000$: The horizontal velocity component u along $x = 0.5L$ (left panel) and the vertical velocity component v along $y = -0.25L$ (right panel) using the Wendland and Laguerre-Gauss kernels with different resolutions and the comparison with the reference obtained by Glowinski et al. [56].

4.5. Lid-driven semi-circular cavity problem

In this section, we explore two-dimensional lid-driven cavity flows characterized by a semi-circular shape, aiming to assess the stability, accuracy, and capability to handle irregular geometries of the proposed kernel. For the boundary conditions, a prescribed velocity of $U_{wall} = 1.0$ is assigned to the upper moving wall, while non-slip wall conditions are applied to the remaining boundaries. The simulation time is set to $t = 30$ and Reynolds number as $Re = 1000$.

Fig. 9 displays the distribution of velocity magnitude obtained with the Wendland (left panel) and Laguerre-Gauss (right panel) kernels at a resolution of $dp = 1/129$. The results reveal that the velocity fields obtained with both kernels exhibit good smoothness. Additionally, as shown in Fig. 10, the obtained velocity profiles indicate considerably improved accuracy and good convergence with the Laguerre-Gauss kernel.

4.6. 2D oscillating plate

In 2D solid dynamics, following Refs. [39,57,58], the oscillating plate with the length $L = 0.2$ m and the thickness $H = 0.02$ m, where one edge is fixed, and the other edges are free, is studied to verify the accuracy of the Laguerre-Gauss kernel. In this case, the density $\rho_0 = 1000.0$ kg/m³, Young's modulus $E = 2$ MPa, physical time $t = 2$ s and Poisson's ratio ν is changeable. Also, the initial velocity is perpendicular to the plate strip given by

$$v_y = v_f c \frac{f(x)}{f(L)}, \quad (34)$$

with v_f denoting a constant and $f(x)$ written as

$$f(x) = (\sin(kL) + \sinh(kL))(\cos(kx) - \cosh(kx)) - (\cos(kL) + \cosh(kL))(\sin(kx) - \sinh(kx)). \quad (35)$$

Here, k is derived by

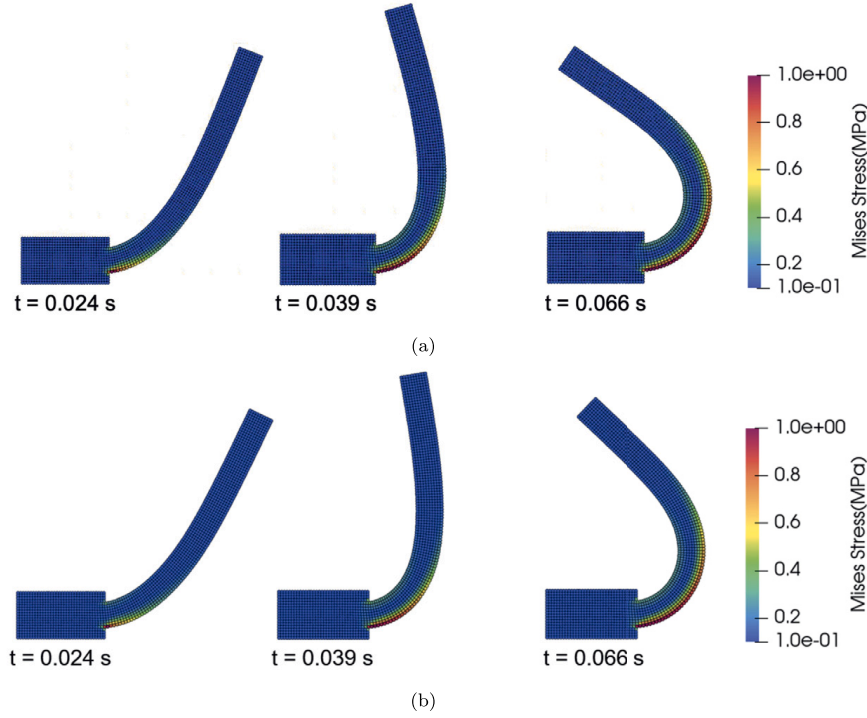


Fig. 11. 2D oscillating plate: The Mises stress contour ranging from 0.1 MPa to 1.0 MPa using the Wendland (a) and Laguerre-Gauss (b) kernels with $\nu_f = 0.15$ and Poisson's ratio $\nu = 0.3975$ under the spatial resolution $H/dp = 10$.

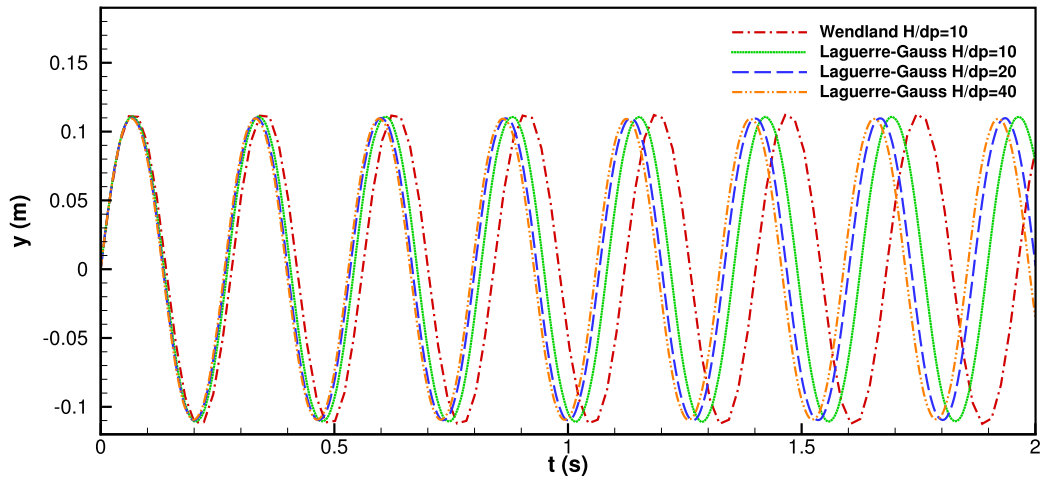


Fig. 12. 2D oscillating plate: The vertical position y located at the midpoint of the plate strip end [39] using the Laguerre-Gauss kernel with the $\nu_f = 0.05$ and Poisson's ratio $\nu = 0.3975$ under the spatial resolution as $H/dp = 10, 20$ and 40 , and the comparison with the result using the Wendland kernel at $H/dp = 10$. In this case, the theoretical oscillation period $T_{\text{Theoretical}} = 0.254$ and the oscillation periods using the truncated Laguerre-Gauss and Wendland kernels are $T_{\text{LG}} = 0.271$ with $\text{Error}_{\text{LG}} = 6.7\%$ and $T_{\text{W}} = 0.282$ with $\text{Error}_{\text{W}} = 11.0\%$, respectively.

$$\cos(kL) \cosh(kL) = -1, \quad (36)$$

with $kL = 1.875$. Also, the theoretical frequency ω is given by

$$\omega^2 = \frac{EH^2k^4}{12\rho(1-\nu^2)}. \quad (37)$$

To keep the results reasonable, the largest spatial resolution is set as $H/dp = 10$ in the case of discretizing the computational domain and the other two spatial resolutions $H/dp = 20$ and 40 are applied for the convergence study [39,57,58].

Fig. 11 presents the Mises stress contour ranging from 0.1 MPa to 1.0 MPa using the Wendland (a) and Laguerre-Gauss (b) kernels with $\nu_f = 0.15$ and Poisson's ratio $\nu = 0.3975$ as well as the resolution $H/dp = 10$, indicating that the results obtained by both kernels are smooth. Also, Fig. 12 shows the vertical position y at the midpoint of the plate strip end using the Laguerre-Gauss kernel among the three resolutions and the comparison with that using the Wendland kernel with $H/dp = 10$. As the resolution increases, the gap between the different resolutions decreases, indicating the results are convergent and achieve second-order convergence. Additionally,

Table 6

2D oscillating plate: The oscillation period using the Wendland and Laguerre-Gauss kernels with various v_f and v under the spatial resolution $H/dp = 10$ and the comparisons with the theoretical results.

v_f	v	$T_{\text{Theoretical}}$	T_W	Error _W	T_{LG}	Error _{LG}
0.05	0.22	0.27009	0.29792	10.3%	0.28824	6.7%
0.1	0.22	0.27009	0.29721	10.0%	0.29016	7.4%
0.05	0.3	0.26412	0.291416	10.3%	0.28184	6.7%
0.1	0.3	0.26412	0.29130	10.3%	0.28138	6.5%
0.05	0.4	0.25376	0.28012	10.4%	0.27072	6.9%
0.1	0.4	0.25376	0.28065	10.6%	0.27134	6.5%

the Laguerre-Gauss kernel demonstrates the ability to attain a more precise oscillation period than the Wendland kernel. However, it is worth noting that the improvement in accuracy achieved is relatively modest compared with previous simulations. This is primarily attributed to the fact that the outcomes obtained by the latter kernel at lower resolutions are already in close proximity to the theoretical oscillation period. For quantitative analysis, two kernels are applied to calculate the oscillation period T separately at low resolution $H/dp = 10$ to verify the performance. Table 6 lists the oscillation period and errors using the Wendland (denoted as T_W and Error_W, respectively) and Laguerre-Gauss (denoted as T_{LG} and Error_{LG}, respectively) kernels with different v_f and v compared with the theoretical oscillation period (denoted as $T_{\text{Theoretical}}$) under the resolution $H/dp = 10$, clearly showing that the errors of using the Laguerre-Gauss kernel are much smaller than that of using the Wendland kernel. Furthermore, the CPU wall-clock times obtained by total Lagrangian SPH using the Wendland and Laguerre-Gauss kernels at the spatial resolution $dp = 1/40$ in the whole process are 537.68 s and 530.25 s, implying that the computational efficiency of using both kernels is comparable.

4.7. 2D colliding rubber rings

In order to assess the stability and accuracy in cases with irregular initial particle distributions in more complex geometries, this section simulates the well-known benchmark case of the collision of two rubber rings, which was first simulated by Swegle et al. [11] and also investigated in Refs [57,59–61]. In this case, two rings, each with an inner radius of 0.03 and an outer radius of 0.04, approach one another at an initial velocity magnitude denoted by v_0 , and a preliminary center-to-center gap of 0.09. The rings are characterized by material properties: a density of $\rho_0 = 1200$, Young's modulus $E = 1 \times 10^7$, and Poisson's ratio $\nu = 0.4$. The setup employs an initial particle spacing of 0.001 and an initial velocity value of $v_0 = 0.06c_0$ and conducts trials using the Wendland and Laguerre-Gauss kernels, respectively.

Fig. 13 depicts the Mises stress contours, generated using the Wendland (a) and Laguerre-Gauss (b) kernels with an initial particle spacing of 0.001 and initial velocity magnitude of $v_0 = 0.06c_0$ at different moments. Although the quantitative comparison is not easy due to the lack of convergent numerical results, one can still observe that the particle distribution, as shown in the inserts of Fig. 13, obtained by the latter kernel is more uniform than that by the former one.

4.8. 3D oscillating plate

In this section, we simulate the oscillation of a 3D thin plate to validate the robustness and accuracy of the truncated Laguerre-Gauss kernel in 3D dimensions. Following Refs. [62–64], a square plate is set as the length $L = 0.4$ m, width $W = 0.4$ m and height $H = 0.01$ m with the density $\rho_0 = 1000.0$ kg/m³, Young's modulus $E = 100.0$ MPa and Poisson's ratio $\nu = 0.3$ and the simulation time is $t = 0.07$ s. Under the condition that the support effect is applied on the centerline of lateral faces, i.e. the corresponding particle is fixed in the z -direction, the initial velocity v_z along the z -direction is given as

$$v_z(x, y) = \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{W}, \quad (38)$$

where m and n control the x - and y - directional vibration modes, respectively. Also, the theoretical vibration period of the plate is given by

$$T = \frac{2}{\pi} \left[\left(\frac{m}{L} \right)^2 + \left(\frac{n}{W} \right)^2 \right]^{-1} \sqrt{\frac{\rho H}{D}}, \quad (39)$$

with the flexural rigidity $D = EH^3 / [12(1 - \nu^2)]$. Also, we apply three spatial resolutions $H/dp = 5, 7$ and 9 in the case.

Fig. 14 presents the Mises stress contour ranging from 0 MPa to 0.28 MPa with the vibration mode of $(m, n) = (2, 1)$ at $t = 0.00279$ s and $(m, n) = (2, 2)$ at $t = 0.00219$ s using the Laguerre-Gauss kernel with the spatial resolution $H/dp = 9$, showing the proposed kernel enables obtaining smooth results of Mises stress. Also, Fig. 15 illustrates the oscillation period of the kinetic energy for the vibration modes $(m, n) = (2, 1)$ and $(2, 2)$ using the Laguerre-Gauss kernel at three different resolutions, the comparison to the results obtained using the Wendland kernel and the kinetic theoretical solution with $H/dp = 5$, shows that the Laguerre-Gauss kernel can obtain higher accuracy. Also, in both vibrational modes, the elastic strain and total energy using the Laguerre-Gauss kernel with $H/dp = 9$ are displayed in Fig. 15, the magnitude and duration of the kinetic energy remain unchanged and the current elastic strain energy, calculated from the deformation gradient, exhibits a slight disparity in amplitude compared to the kinetic energy, as anticipated [39].

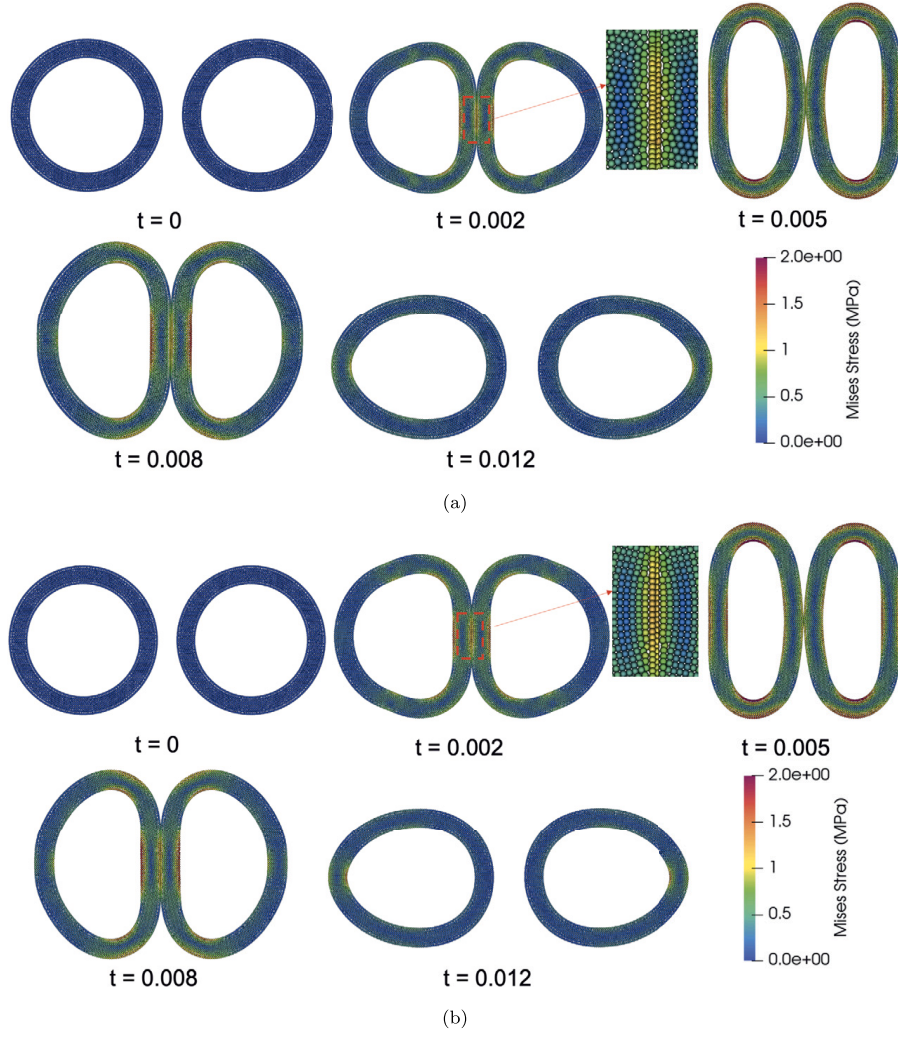


Fig. 13. 2D colliding rubber rings: The Mises stress contour ranging from 0 MPa to 2.0 MPa using the Wendland (a) and Laguerre-Gauss (b) kernels with the initial particle spacing of 0.001 and initial velocity magnitude $v_0 = 0.06c_0$ at various time instants $t = 0, 0.002, 0.005, 0.008$ and 0.012 .

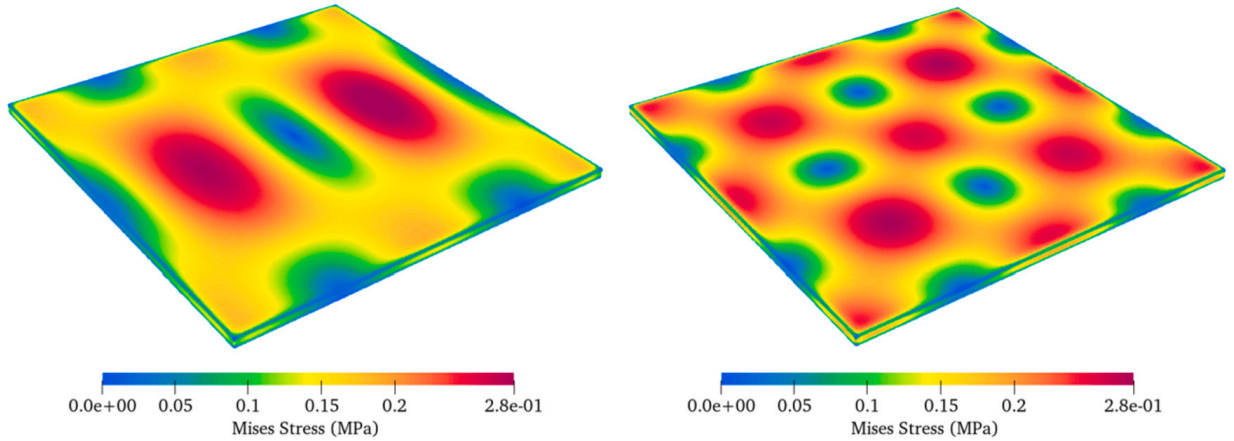


Fig. 14. 3D oscillating plate: The Mises stress contour ranging from 0 MPa to 0.28 MPa with the vibration mode of $(m, n) = (2, 1)$ at $t = 0.00279$ s (left panel) and $(m, n) = (2, 2)$ at $t = 0.00219$ s (right panel) using the Laguerre-Gauss kernel with the spatial resolution $H/dp = 9$.

This confirms that quite good energy-conservation property [63,64] is obtained using the proposed kernel. Besides, it is observed that with the increase in spatial resolutions, the results are convergent. The wall-clock time for simulation of the vibration mode of $(m, n) = (2, 1)$ obtained by total Lagrangian SPH using the Wendland and Laguerre-Gauss kernels with spatial resolution $H/dp = 9$ in entire process are approximately 21910.9 s and 21906.0 s, respectively, showing that the computational efficiency of using the Laguerre-Gauss kernel is at the same level as that of using the Wendland kernel.

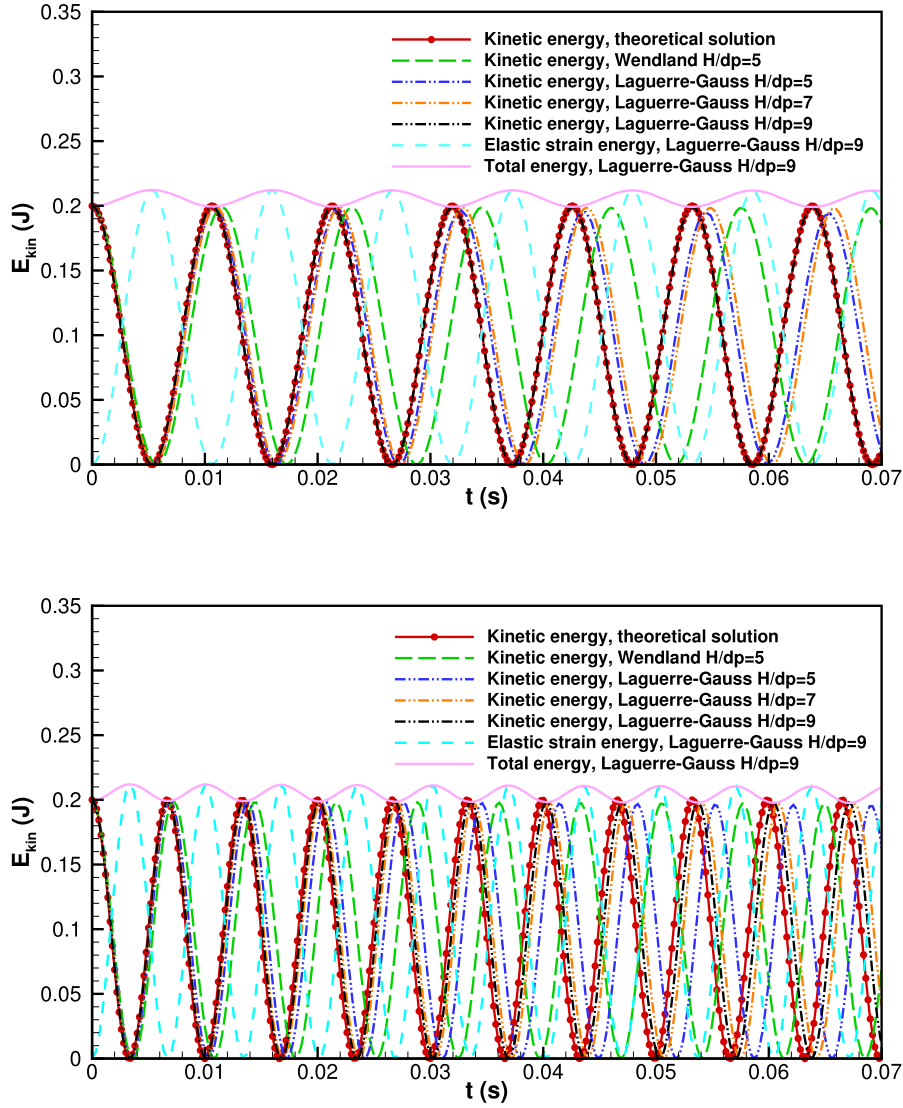


Fig. 15. 3D oscillating plate: The oscillation period of the kinetic energy for the vibration mode of $(m, n) = (2, 1)$ (upper panel) and $(2, 2)$ (bottom panel) using the Laguerre-Gauss kernel with the spatial resolution as $H/dp = 5, 7$ and 9 , and the comparisons with the results using the Wendland kernel and the kinetic theoretical solutions with $H/dp = 5$. Besides, the elastic strain energy and total energy obtained by the Laguerre-Gauss kernel are also displayed with $H/dp = 9$.

4.9. 3D bending column

We also consider a bending-dominated problem called bending column to investigate the robustness and accuracy of the Laguerre-Gauss kernel further. Following Refs. [65,39], a rubber-like material with a length of $L = 6$ m, a height of $H = 1$ m and a square cross-sectional area which is fixed at the bottom is applied in the numerical simulations. At the same time, the initial velocity conditions are given as $\mathbf{v}_0 = 10(\frac{\sqrt{3}}{2}, \frac{1}{2}, 0)^T$, density $\rho_0 = 1100$ kg/m³, Young's modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$. Also, three spatial resolutions, including $dp = 1/6$, $dp = 1/12$ and $dp = 1/24$, are applied for the convergence study.

Fig. 16 displays the Mises stress contour, generated through the Wendland (a) and Laguerre-Gauss (b) kernels with the resolution $H/dp = 12$ at different time instants. This illustrates that the stress distribution is smooth when applying the proposed kernels, and the outcomes from both kernels exhibit consistency. Besides, Fig. 17 portrays the vertical position z observed at the top of the column [39] using the Laguerre-Gauss kernel at three different resolutions $dp = 1/6, 1/12$ and $1/24$, and their comparisons with the Wendland kernel at the resolution $dp = 1/6$ as well as the result obtained by Aguirre et al. [66], showing that the Laguerre-Gauss kernel can improve accuracy significantly compared to the Wendland kernel as expected and the results converge rapidly with the increase of the spatial resolutions. Also, Fig. 18 illustrates the z -axis velocity at the top of the column [67], utilizing the Wendland and Laguerre-Gauss kernels across three distinct resolutions. The observation confirms that both kernels yield consistent results at higher resolutions $dp = 1/24$. Moreover, at lower resolutions $dp = 1/6$, the velocity profile obtained using the Laguerre-Gauss kernel is significantly more accurate than that using the Wendland kernel.

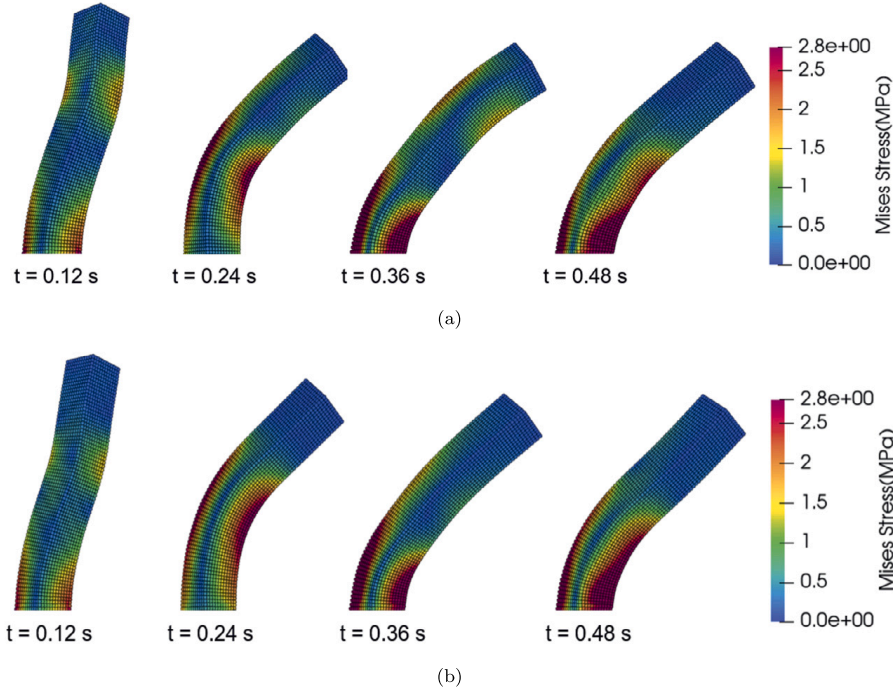


Fig. 16. 3D Bending column: The Mises stress contour ranging from 0 MPa to 2.8 MPa using the Wendland (a) and Laguerre-Gauss (b) kernels with the spatial resolution $H/dp = 12$ at different time instants.

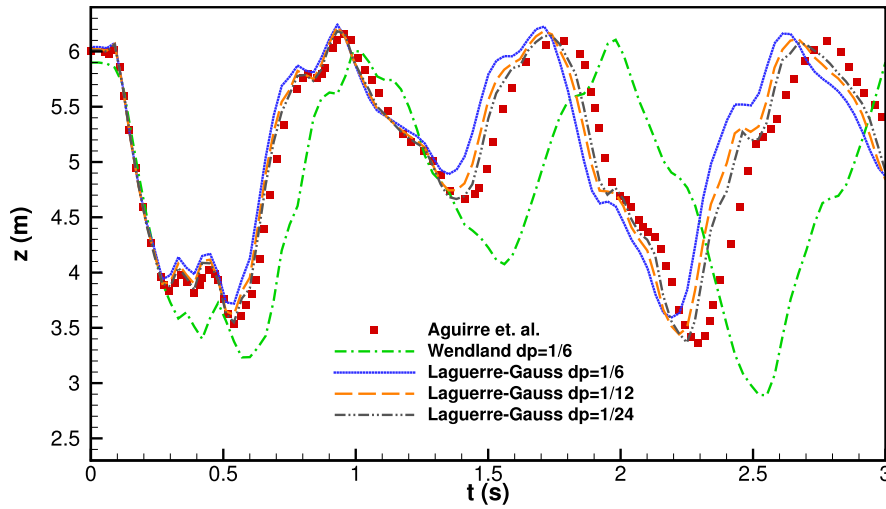


Fig. 17. 3D Bending column: The vertical position z located at the top of the column [39] using the Wendland and Laguerre-Gauss kernels with three different resolutions and the comparisons with the reference data from Aguirre et al. [66].

5. Summary and conclusion

In this paper, the error of the truncated Laguerre-Gauss kernel is analyzed, and the results show that the proposed kernel introduces much less truncation error and relaxation residue after particle relaxation than the Wendland and Laguerre-Wendland kernels, indicating its considerably improved accuracy and stability. Furthermore, a set of 2D and 3D numerical examples for fluid dynamics and solid dynamics are tested and have demonstrated that using the proposed kernel enables obtaining considerably higher accuracy than the Wendland kernel with comparable computational efficiency. Note that although we here focus on the 4th-order Laguerre-Wendland and Laguerre-Gauss kernels, it is also interesting to study whether other high-order kernels, such as those proposed in Refs. [20,14,15,22], have similar properties as future work. Also note that the truncated Laguerre-Gauss kernel studied here is limited in the Eulerian SPH method for fluids and total Lagrangian SPH method for solids, as a choice for general numerical discretization, it can be applied for more complex problems, such as multi-phase flows. However, its applicability and numerical stability within the updated Lagrangian framework still necessitates further investigation.

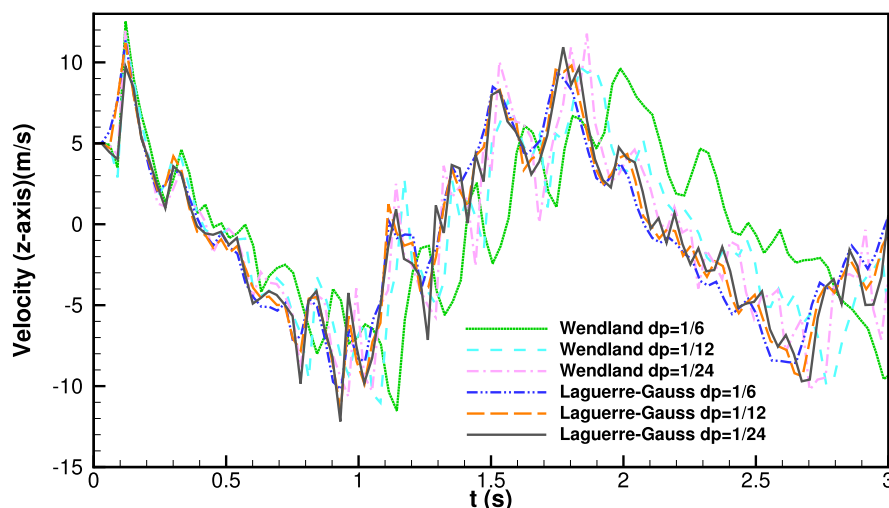


Fig. 18. 3D Bending column: The velocity along the z-axis at the top of the column [67] using the Wendland and Laguerre-Gauss kernels with three different resolutions.

CRediT authorship contribution statement

Zhentong Wang: Writing – original draft, Visualization, Methodology, Investigation, Conceptualization. **Bo Zhang:** Writing – review & editing, Investigation. **Oskar J. Haidn:** Writing – review & editing, Supervision, Investigation. **Xiangyu Hu:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code is open source.

References

- [1] S. Litvinov, X. Hu, N.A. Adams, Towards consistence and convergence of conservative SPH approximations, *J. Comput. Phys.* 301 (2015) 394–401.
- [2] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (1977) 1013–1024.
- [3] J.J. Monaghan, Simulating free surface flows with SPH, *J. Comput. Phys.* 110 (2) (1994) 399–406.
- [4] L.D. Libersky, A.G. Petschek, T.C. Carney, J.R. Hipp, F.A. Allahdadi, High strain Lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response, *J. Comput. Phys.* 109 (1) (1993) 67–75.
- [5] P. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, *Comput. Methods Appl. Mech. Eng.* 139 (1–4) (1996) 375–408.
- [6] S.M. Longshaw, B.D. Rogers, Automotive fuel cell sloshing under temporally and spatially varying high acceleration using GPU-based Smoothed Particle Hydrodynamics (SPH), *Adv. Eng. Softw.* 83 (2015) 31–44.
- [7] H. Gotoh, A. Khayyer, Y. Shimizu, Entirely Lagrangian meshfree computational methods for hydroelastic fluid-structure interactions in ocean engineering—reliability, adaptivity and generality, *Appl. Ocean Res.* 115 (2021) 102822.
- [8] Y. Sun, G. Xi, Z. Sun, A generic smoothed wall boundary in multi-resolution particle method for fluid–structure interaction problem, *Comput. Methods Appl. Mech. Eng.* 378 (2021) 113726.
- [9] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (3) (1977) 375–389.
- [10] M. Liu, G. Liu, Smoothed particle hydrodynamics (SPH): an overview and recent developments, *Arch. Comput. Methods Eng.* 17 (2010) 25–76.
- [11] J.W. Swegle, D.L. Hicks, S.W. Attaway, Smoothed particle hydrodynamics stability analysis, *J. Comput. Phys.* 116 (1) (1995) 123–134.
- [12] J.P. Morris, *Analysis of Smoothed Particle Hydrodynamics with Applications*, Monash University Australia, 1996.
- [13] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *Int. J. Numer. Methods Eng.* 66 (13) (2006) 2064–2085.
- [14] S.J. Lind, P. Stansby, High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion, *J. Comput. Phys.* 326 (2016) 290–311.
- [15] G. Fourtakas, P. Stansby, B.D. Rogers, S. Lind, An Eulerian–Lagrangian incompressible SPH formulation (ELI-SPH) connected with a sharp interface, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 532–552.
- [16] H.H. Bui, G.D. Nguyen, Smoothed particle hydrodynamics (SPH) and its applications in geomechanics: from solid fracture to granular behaviour and multiphase flows in porous media, *Comput. Geotech.* 138 (2021) 104315.
- [17] A. Khayyer, H. Gotoh, Y. Shimizu, On systematic development of FSI solvers in the context of particle methods, *J. Hydrodyn.* 34 (3) (2022) 395–407.
- [18] Y. Shimizu, A. Khayyer, H. Gotoh, An SPH-based fully-Lagrangian meshfree implicit FSI solver with high-order discretization terms, *Eng. Anal. Bound. Elem.* 137 (2022) 160–181.
- [19] G.R. Johnson, S.R. Beissel, Normalized smoothing functions for SPH impact computations, *Int. J. Numer. Methods Eng.* 39 (16) (1996) 2725–2741.
- [20] S. Sibilla, An algorithm to improve consistency in smoothed particle hydrodynamics, *Comput. Fluids* 118 (2015) 148–158.

- [21] V. Zago, L.J. Schulze, G. Bilotta, N. Almashan, R. Dalrymple, Overcoming excessive numerical dissipation in SPH modeling of water waves, *Coast. Eng.* 170 (2021) 104018.
- [22] A.M. Nasar, G. Fourtakas, S.J. Lind, J. King, B.D. Rogers, P.K. Stansby, High-order consistent SPH with the pressure projection method in 2-D and 3-D, *J. Comput. Phys.* 444 (2021) 110563.
- [23] A. Vergnaud, G. Oger, D. Le Touzé, Investigations on a high order SPH scheme using WENO reconstruction, *J. Comput. Phys.* 477 (2023) 111889.
- [24] C. Zhang, G. Xiang, B. Wang, X. Hu, N.A. Adams, A weakly compressible SPH method with WENO reconstruction, *J. Comput. Phys.* 392 (2019) 1–18.
- [25] Z. Wang, C. Zhang, O.J. Haidn, X. Hu, An Eulerian SPH method with WENO reconstruction for compressible and incompressible flows, *J. Hydrodyn.* (2023) 1–12.
- [26] W.-K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (I) methodology and convergence, *Comput. Methods Appl. Mech. Eng.* 143 (1–2) (1997) 113–154.
- [27] G.A. Dilts, Moving-least-squares-particle hydrodynamics—I. Consistency and stability, *Int. J. Numer. Methods Eng.* 44 (8) (1999) 1115–1155.
- [28] G.A. Dilts, Moving least-squares particle hydrodynamics II: conservation and boundaries, *Int. J. Numer. Methods Eng.* 48 (10) (2000) 1503–1524.
- [29] G. Bilotta, G. Russo, A. Hérault, C. Del Negro, et al., Moving least-squares corrections for smoothed particle hydrodynamics, *Ann. Geophys.* (2011).
- [30] H. Jiang, Y. Chen, X. Zheng, S. Jin, Q. Ma, A study on stable regularized moving least-squares interpolation and coupled with SPH method, *Math. Probl. Eng.* 2020 (2020).
- [31] J. Ha, A numerical study of the application of radial basis function and generalised smoothed particle hydrodynamics to CFD, in: 15th Australasian Fluid Mechanics Conference, The University of Sydney, Sydney, Australia, 2004.
- [32] R. Brownlee, P. Houston, J. Levesley, S. Rosswog, Enhancing SPH using moving least-squares and radial basis functions, in: *Algorithms for Approximation: Proceedings of the 5th International Conference*, July 2005, Springer, Chester, 2007, pp. 103–112.
- [33] A.Y. Ghoneim, A smoothed particle hydrodynamics-phase field method with radial basis functions and moving least squares for meshfree simulation of dendritic solidification, *Appl. Math. Model.* 77 (2020) 1704–1741.
- [34] C. Zhang, M. Rezavand, Y. Zhu, Y. Yu, D. Wu, W. Zhang, J. Wang, X. Hu, SPHInXsys: an open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics, *Comput. Phys. Commun.* 267 (2021) 108066.
- [35] C. Zhang, Y.-j. Zhu, D. Wu, N.A. Adams, X. Hu, Smoothed particle hydrodynamics: methodology development and recent achievement, *J. Hydrodyn.* 34 (5) (2022) 767–805.
- [36] X. Yang, S. Peng, M. Liu, A new kernel function for SPH with applications to free surface flows, *Appl. Math. Model.* 38 (15–16) (2014) 3822–3833.
- [37] Y. Zhu, C. Zhang, Y. Yu, X. Hu, A CAD-compatible body-fitted particle generator for arbitrarily complex geometry and its application to wave-structure interaction, *J. Hydrodyn.* 33 (2) (2021) 195–206.
- [38] Y. Ren, P. Lin, C. Zhang, X. Hu, An efficient correction method in Riemann SPH for the simulation of general free surface flows, *Comput. Methods Appl. Mech. Eng.* 417 (2023) 116460.
- [39] D. Wu, C. Zhang, X. Tang, X. Hu, An essentially non-hourglass formulation for total Lagrangian smoothed particle hydrodynamics, *Comput. Methods Appl. Mech. Eng.* 407 (2023) 115915.
- [40] J. Vila, On particle weighted methods and smooth particle hydrodynamics, *Math. Models Methods Appl. Sci.* 9 (02) (1999) 161–209.
- [41] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [42] E.F. Toro, The hllc Riemann solver, *Shock Waves* 29 (8) (2019) 1065–1082.
- [43] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*, Springer Science & Business Media, 2013.
- [44] C. Zhang, X. Hu, N.A. Adams, A weakly compressible SPH method based on a low-dissipation Riemann solver, *J. Comput. Phys.* 335 (2017) 605–620.
- [45] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.* 4 (1995) 389–396.
- [46] Y. Zhu, X. Hu, An L2-norm regularized incremental-stencil WENO scheme for compressible flows, *Comput. Fluids* 213 (2020) 104721.
- [47] V.K. Chakravarthy, D. Chakraborty, Damping numerical oscillations in hybrid solvers through detection of Gibbs phenomenon, *Int. J. Numer. Methods Fluids* 84 (12) (2017) 699–714.
- [48] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.
- [49] G.I. Taylor, A.E. Green, Mechanism of the production of small eddies from large ones, *Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci.* 158 (895) (1937) 499–521.
- [50] F.M. White, J. Majdalani, *Viscous Fluid Flow*, vol. 3, McGraw-Hill, New York, 2006.
- [51] T. Khademinezhad, P. Talebizadeh, H. Rahimzadeh, Numerical study of unsteady flow around a square cylinder in compare with circular cylinder, in: *Conference Paper*, 2015.
- [52] P.-H. Chiu, R.-K. Lin, T.W. Sheu, A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries, *J. Comput. Phys.* 229 (12) (2010) 4476–4500.
- [53] D.-V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.* 220 (1) (2006) 109–138.
- [54] C. Brehm, C. Hader, H.F. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 295 (2015) 475–504.
- [55] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1) (1998) 35–57.
- [56] R. Glowinski, G. Guidoboni, T.-W. Pan, Wall-driven incompressible viscous flow in a two-dimensional semi-circular cavity, *J. Comput. Phys.* 216 (1) (2006) 76–91.
- [57] J.P. Gray, J.J. Monaghan, R. Swift, SPH elastic dynamics, *Comput. Methods Appl. Mech. Eng.* 190 (49–50) (2001) 6641–6662.
- [58] L.D. Landau, *Course of Theoretical Physics, Theory of Elasticity*, vol. 10, 1986, pp. 32–35.
- [59] J.J. Monaghan, SPH without a tensile instability, *J. Comput. Phys.* 159 (2) (2000) 290–311.
- [60] C. Zhang, X.Y. Hu, N.A. Adams, A generalized transport-velocity formulation for smoothed particle hydrodynamics, *J. Comput. Phys.* 337 (2017) 216–232.
- [61] S. Zhang, S.D. Lourenço, D. Wu, C. Zhang, X. Hu, Essentially non-hourglass and non-tensile-instability SPH elastic dynamics, *arXiv preprint*, arXiv:2310.08444, 2023.
- [62] A.W. Leissa, *Vibration of Plates* vol. 160, Scientific and Technical Information Division, National Aeronautics and Space Administration, 1969, pp. 1651–1663.
- [63] A. Khayyer, H. Gotoh, Y. Shimizu, Y. Nishijima, A 3D Lagrangian meshfree projection-based solver for hydroelastic fluid-structure interactions, *J. Fluids Struct.* 105 (2021) 103342.
- [64] A. Khayyer, Y. Shimizu, H. Gotoh, S. Hattori, A 3D SPH-based entirely Lagrangian meshfree hydroelastic FSI solver for anisotropic composite structures, *Appl. Math. Model.* 112 (2022) 560–613.
- [65] C. Zhang, J. Wang, M. Rezavand, D. Wu, X. Hu, An integrative smoothed particle hydrodynamics method for modeling cardiac function, *Comput. Methods Appl. Mech. Eng.* 381 (2021) 113847.
- [66] M. Aguirre, A.J. Gil, J. Bonet, A.A. Carreno, A vertex centred finite volume Jameson–Schmidt–Turkel (JST) algorithm for a mixed conservation formulation in solid dynamics, *J. Comput. Phys.* 259 (2014) 672–699.
- [67] C. Zhang, Y. Zhu, Y. Yu, D. Wu, M. Rezavand, S. Shao, X. Hu, An artificial damping method for total Lagrangian SPH method with application in biomechanics, *Eng. Anal. Bound. Elem.* 143 (2022) 1–13.