

# Advancing consistency and efficiency in smoothed particle hydrodynamics

Bo Zhang

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. Christoph Meier  
Prüfer der Dissertation: 1. Priv.-Doz. Dr.-Ing. Xiangyu Hu  
2. Prof. Dr.-Ing. Oskar J. Haidn

Die Dissertation wurde am 11.11.2024 bei der Technischen Universität München  
eingereicht und durch die TUM School of Engineering and Design am 04.03.2025  
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 at this or any other university. This dissertation is my own work, except where indicated in the text and Acknowledgments, and contains no work done in collaboration with others.

Bo Zhang  
April 17, 2025

© Bo Zhang, 2025  
bo.zhang.aer@tum.de

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

Released April 17, 2025  
Typesetting  $\LaTeX$



*All the roads I've traveled have led me to the bright, where I can see myself.  
I am grateful to my parents and everyone I've met along the way.*

*Bo Zhang*



# Abstract

This cumulative thesis is devoted to advancing the consistency and efficiency of the smoothed particle hydrodynamics (SPH) method for multi-physics simulations and complex engineering applications, addressing issues on numerical consistency, computational efficiency, and the development of a consistent and efficient scientific computing library. In particular, a reverse kernel gradient correction (RKGC) formulation is proposed for conservative SPH approximations to achieve high-order consistency in both Lagrangian and Eulerian SPH methods. The splitting SPH method is employed to solve PDE-constrained optimization problems from an “all-at-once” perspective, leading to significantly improved optimization efficiency. Additionally, an automated regression testing method is introduced to ensure consistency in the development and release of the SPH-based open-source scientific computing library, supporting efficient development and application of SPH algorithms.

The first part of this thesis addresses the critical challenge of achieving high-order consistency in SPH while preserving its conservative properties. Traditional SPH formulations often struggle to balance discrete conservation with consistency, particularly when applying higher-order corrections such as kernel gradient correction (KGC). This work introduces the RKGC formulation, a novel approach that integrates conservation with first-order consistency by relaxing particles based on the KGC matrix. Extensive numerical tests demonstrate significant improvements in numerical accuracy for both Lagrangian and Eulerian SPH methods, even with a notably reduced ratio of smoothing length to particle spacing. Particularly, this method effectively mitigates long-standing high dissipation issues in simulations of free-surface flow problems. This advancement not only enhances accuracy but also provides a foundation for potential extensions to higher-order consistency, opening up new possibilities for more precise SPH simulations in engineering applications.

The second part of this thesis focuses on applying the SPH method to PDE-constrained optimization problems, specifically optimizing heat conduction through the efficient distribution of high thermal conductivity materials, which is crucial for electronic cooling systems and other thermal management applications. We develop a target-driven, all-at-once optimization method that employs a splitting SPH technique, breaking the optimization process into several simple, manageable steps. This approach eliminates the traditional reliance on adjoint state equations and fully converged state fields in each iteration, significantly accelerating the optimization process. The proposed framework effectively balances target attainment with minimal computational overhead. Additionally, by incorporating implicit SPH splitting operators and numerical regularization techniques, we achieve faster information propagation and improved numerical stability. This method is validated through a series of representative problems, demonstrating superior computational efficiency and yielding optimal thermal conductivity distributions that are easier to implement with available materials, making it highly practical for real-world applications.

The third part of this thesis addresses the growing complexity in developing and maintaining scientific computing libraries for multi-physics simulations, with

a focus on methods applicable to SPH-based libraries. This work presents the design and implementation of an automated regression testing framework for scientific computing libraries, using SPHinXsys, an open-source SPH-based library, as an illustrative example. The framework generates a comprehensive reference database for each benchmark test, capturing data from multiple test executions to account for uncertainties arising from parallel computing, particle relaxation, physical instabilities, and other factors. This reference database enables curve-similarity comparisons between new and baseline results following code modifications, providing a robust assessment of current result validity. The automated regression test system runs tests automatically whenever code updates occur, covering a wide range of multi-physics problems, including fluid dynamics, solid dynamics, thermal and mass diffusion, and their coupling, to ensure that accuracy and functionality of the code remain. The flexibility of this framework allows it to be adapted for other scientific computing libraries, promoting consistent and reliable software development practices in computational science while enhancing development efficiency.

Collectively, these studies significantly advance the consistency and efficiency of the SPH method, addressing not only its numerical fundamentals and optimization applications but also the reliability and efficiency of corresponding software development. This work provides a comprehensive foundation for enhancing the performance, reliability, and applicability of the SPH method across a variety of engineering and scientific research domains.

# Acknowledgements

To begin, I would like to express my deep and sincere gratitude to my doctoral supervisor, PD Dr. Xiangyu Hu, for his unwavering support, careful guidance, and encouragement throughout my Ph.D. journey. His expertise in numerical methods and profound knowledge of fluid dynamics have been invaluable to my research. In every challenging moment, Dr. Hu was always available to help, engaging actively in discussions and offering insightful solutions. His steady encouragement and guidance reignited my motivation when I felt anxious about my progress. Each deep discussion with Dr. Hu has been a meaningful experience, contributing immensely to my growth as an excellent independent researcher.

I want to extend my deepest gratitude to my master's supervisor, Dr. Xiaoqing Qiang, at Shanghai Jiao Tong University. Our relationship transcends that of a supervisor and student; we are friends. His unwavering academic and personal support has been a tremendous source of strength, and I am genuinely grateful for his encouragement. I am equally thankful to my bachelor's supervisor, Prof. Jianfeng Zhu, at Xiamen University. His guidance instilled in me the importance of thoughtful decision-making, a principle that has been a compass for me throughout my studies and life. His encouragement to deepen my scientific understanding has continually inspired me to strive for excellence in my academic work and research.

It has been a joy and privilege to collaborate with my colleagues at the Chair of Aerodynamics and Fluid Mechanics at the Technical University of Munich, including Dr. Chi Zhang, Dr. Massoud Rezavand, Dr. Yujie Zhu, Dong Wu, Shuoguo Zhang, and Feng Wang. Working together has always been both productive and fulfilling. My special thanks go to Prof. Nikolaus A. Adams and Prof. Christian Stemmer, as well as our dedicated secretaries, Hua Liu and Angela Grygier, for their invaluable support and kindness. I am also deeply thankful to my friends from other research institutions, including Yongchuan Yu, Xiaojing Tang, Zhentong Wang, Mai Ye, Chenxi Zhao, Junwei Jiang, etc., for their companionship and encouragement during this memorable journey.

Above all, my deepest gratitude goes to my parents. Their unwavering support and respect for my decisions have been the foundation of my strength. Our weekly conversations have been a constant source of comfort, reminding me to move forward with courage, pursue my passions, and believe in myself. I could not have reached this milestone without their love and encouragement. As I reflect on this journey, I am grateful for the companionship that has illuminated my path, turning challenges into stepping stones and triumphs into shared joys. With deep appreciation, I look forward to the road ahead, knowing that our bonds will endure as we continue to inspire one another in our pursuit of knowledge and excellence.



# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Numerical methods in scientific computing . . . . .	1
1.2 Review on SPH method . . . . .	2
1.2.1 SPH and its consistency . . . . .	2
1.2.2 SPH in PDEs-constrained optimization . . . . .	4
1.2.3 SPH-based libraries and verification . . . . .	4
1.3 Aims and objectives . . . . .	5
1.4 Outline . . . . .	6
<b>2 Methodology</b>	<b>7</b>
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 Consistency correction and variational analysis . . . . .	9
2.3 SPH discretization . . . . .	13
2.3.1 Weakly compressible SPH (WCSPH) . . . . .	13
2.3.2 Eulerian SPH (ESPH) . . . . .	14
2.3.3 Heat conduction . . . . .	14
2.4 Zero-order consistency satisfaction . . . . .	15
2.4.1 KGC-based particle relaxation . . . . .	15
2.4.2 KGC-based transport-velocity formulation . . . . .	16
2.5 Splitting SPH optimization method . . . . .	16
2.5.1 Implicit scheme . . . . .	16
2.5.2 Evolution of design variable . . . . .	18
2.5.3 Numerical regularization . . . . .	19
2.6 Regression test method . . . . .	19
2.6.1 Time-averaged strategy . . . . .	19
2.6.2 Ensemble-averaged strategy . . . . .	20
2.6.3 Dynamic time warping (DTW) strategy . . . . .	21
2.7 Summary . . . . .	21
<b>3 Summaries of publications</b>	<b>23</b>
3.1 Towards high-order consistency and convergence of conservative SPH approximations . . . . .	23
3.1.1 Summary of the publication . . . . .	23
3.1.2 Individual contributions of the candidate . . . . .	24

3.2	Target-driven splitting SPH optimization of thermal conductivity distribution . . . . .	25
3.2.1	Summary of the publication . . . . .	25
3.2.2	Individual contributions of the candidate . . . . .	26
3.3	Automated regression test method for scientific computing libraries: Illustration with SPHinXsys . . . . .	27
3.3.1	Summary of the publication . . . . .	27
3.3.2	Individual contributions of the candidate . . . . .	28
<b>4</b>	<b>Discussion and outlook</b>	<b>29</b>
4.1	Discussions . . . . .	29
4.2	Outlooks . . . . .	31
	<b>Bibliography</b>	<b>33</b>
<b>A</b>	<b>Original journal papers</b>	<b>43</b>
A.1	Paper I . . . . .	45
A.2	Paper II . . . . .	83
A.3	Paper III . . . . .	99

## Chapter 1

# Introduction

### 1.1 Numerical methods in scientific computing

The evolution of computers has elevated scientific computing to an indispensable tool, playing a pivotal role in solving complex problems across various technologies and industries. These applications include assessing climate change [1, 2], designing new energy conductors [3, 4], improving medical treatments and drug delivery systems in biomedical engineering [5, 6], optimizing aircraft performance and safety in aerospace [7, 8], and enhancing crash safety and vehicle efficiency in automotive engineering [9, 10], among others. Scientific computing is crucial for modeling and predicting systems governed by partial differential equations (PDEs), which describe various physical phenomena such as fluid flow, heat transfer, and structural deformations. Through the numerical simulation of these processes, which are challenging to address using purely analytical or experimental approaches, scientific computing enhances our ability to predict and comprehend both natural phenomena and engineered systems.

Beyond simulation, scientific computing has also become a key tool for solving optimization problems, such as those constrained by PDEs [11]. Many engineering problems involve optimizing certain properties or parameters within a system, subject to physical laws described by PDEs. For example, efficiently enhancing heat conduction through the optimized distribution of a limited quantity of high thermal conductivity material while adhering to the governing equations of heat conduction is significant for cooling electronic devices and numerous other applications [12, 13, 14]. Similarly, in aerodynamics, optimizing the shape of an aircraft wing requires solving fluid dynamics equations to minimize drag or maximize lift, which is essential for enhancing aircraft performance and designing new profiles [15, 16, 17]. PDE-constrained optimization allows us to find feasible solutions to these problems within physical constraints and optimized for performance.

However, modeling, simulating, and optimizing these problems is generally challenging due to their intrinsic complexity, requiring careful consideration of both computational consistency and efficiency. Consequently, numerous numerical methods have been proposed and developed for simulation and optimization, each with different features, typically categorized into mesh-based and mesh-free methods. Moreover, many corresponding numerical algorithms have been developed to improve consistency and efficiency, aiming to make these methods broadly applicable to a wide range of engineering applications with high predictive accuracy. Traditional mesh-based methods, such as the finite difference method (FDM) [18], finite element method (FEM) [19], and finite volume method (FVM) [20], have long been used to discretize and solve PDEs, achieving success in various fields. However, due

to the limitations of mesh-based methods in handling complex geometries, large deformations, and free surfaces, mesh-free methods such as smoothed particle hydrodynamics (SPH) [21, 22], the discrete element method (DEM) [23], and the moving particle semi-implicit method (MPS) [24] have gained traction and played significant roles in addressing these complex problems in scientific computing.

In addition to numerical methods, optimization techniques also play a vital role in effectively addressing complex problems. Optimization methods in this field can be broadly categorized into simulation-based and all-at-once approaches, depending on how the PDE constraint is managed [25, 26]. In simulation-based optimization, the PDE constraint is eliminated by obtaining a converged physical solution using existing solvers. The optimization process then computes the gradients of state variables with respect to design variables at the hypersurface of the physical solution to determine the search direction in each iteration, relying on repeated simulations to evaluate the performance of different design parameters. While advancements in high-performance computing and parallelization have made it feasible to tackle large-scale optimization problems, this approach can still be computationally expensive. In contrast, all-at-once approaches explicitly maintain PDE constraints as part of the optimization target, treating both the state and design variables equally. By embedding the PDEs directly into the optimization process, a clear advantage is that they do not require repeated PDE solutions and solve problems more efficiently because they satisfy the PDE constraint only upon the termination of the optimization.

Whether in simulation or optimization, developing scientific computing software and libraries has been crucial for advancing both the adoption and refinement of these methods. Both commercial and open-source tools have made significant contributions across various fields. Commercial software, such as ANSYS [27] and COMSOL Multiphysics [28], predominantly use mesh-based methods and are widely adopted in the industry for their comprehensive capabilities in simulating complex physical phenomena. Similarly, open-source libraries like OpenFOAM [29] (mesh-based) and SPHinXsys [30] (mesh-free) offer flexible and powerful developing platforms for researchers and engineers. These libraries not only facilitate the simulation of complex systems governed by PDEs but also provide a foundation for integrating optimization algorithms, making them invaluable in both academic research and industrial applications. A key challenge in the continuous evolution of such libraries is ensuring they consistently produce reliable results while maintaining computational efficiency. Achieving this requires a rigorous process of validation and verification that is closely integrated with development and usage [31].

## 1.2 Review on SPH method

Among these numerical methods, SPH has gained increasing attention due to its inherent capability to track moving characteristics and handle large deformations. As the main focus of this thesis, we briefly review the theory and consistency of the SPH method, its applications, and SPH-based computing libraries; for more comprehensive information, readers may refer to review articles [32, 33, 34].

### 1.2.1 SPH and its consistency

As a mesh-free method, SPH, initially proposed by Lucy [21] and Gingold & Monaghan [35] for astrophysical applications, has demonstrated significant success across

a wide range of scientific problems, including fluid dynamics [36, 37, 38], solid dynamics [39, 40, 41], and fluid-structure interaction [33, 42, 43], among others. The SPH method represents the computational domain with a collection of arbitrarily distributed particles, each possessing individual properties, which inherently defines its mesh-free characteristic. The SPH approximation operates on the principle of reconstructing the continuous field and its spatial derivatives from this set of discrete particles through a Gaussian-like smoothing kernel function with compact support [44, 45], as illustrated in Fig. 1.1. The SPH approximation involves

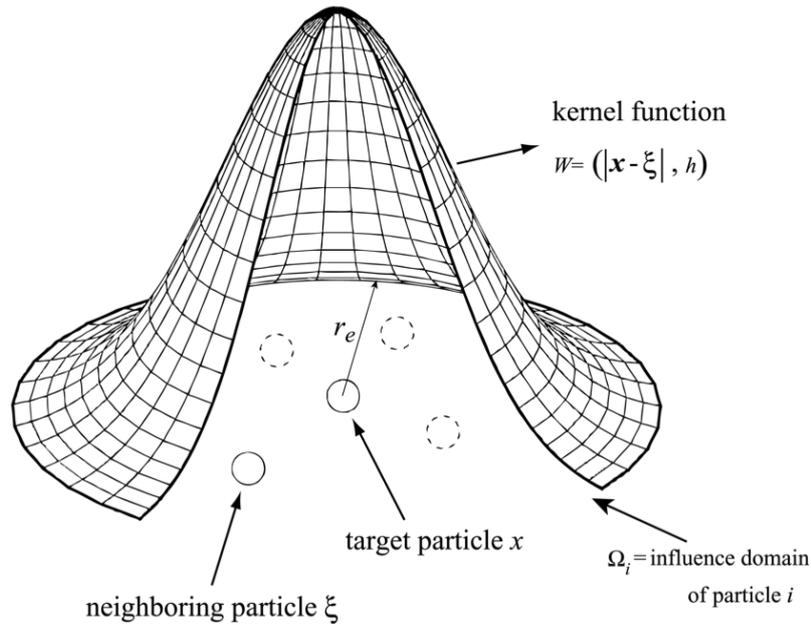


FIGURE 1.1: Particle interaction and the kernel function.

two main parts and generally encounters two types of errors that contribute to the overall truncation error [46, 47]. The first part is the kernel approximation, where the field function is expressed in integral form using a smoothing kernel function within its support domain. This introduces a smoothing error determined by the kernel function, where the leading moments vanish, and the error arises from discrepancies between the smoothing approximations and the exact values. The second part is the particle approximation, where the integral in the kernel approximation is replaced by the summation of the values of adjacent particles within the support domain. This introduces an integration error characterized by the non-vanishing leading moments due to the particle approximation.

For typical SPH kernel functions, such as the cubic B-spline [48] and the Wendland kernel [49], which have second-order accuracy, only the first moment vanishes, corresponding to first-order consistency. For a given kernel function, Quinlan et al. [46] observed that the overall truncation error generally decreases, or shows consistency, with increased resolution only when  $h/\Delta x$ , the ratio between the smoothing length  $h$  and the particle spacing  $\Delta x$ , is large, suggesting a sufficiently small integration error. However, achieving this condition requires excessive particles within the kernel's compact support, resulting in extremely high computational costs. Thus, various approaches have been proposed to minimize integration error or improve consistency with a computationally feasible value of  $h/\Delta x$ , typically less than 1.5 [50]. For the non-conservative form of SPH approximation, where particle-pair differences are used, higher-order consistency can be attained through various

methods [51, 52, 53, 54, 55], though this comes at the expense of computational efficiency. However, applying these methods to the discretization of physical conservation laws still presents the major challenge of non-conservation. Therefore, the conservative form, which uses particle-pair averaging, is employed to ensure conservative discretization with an anti-symmetric formulation between particle pairs. However, achieving high-order consistency in this conservative formulation remains an open challenge. Although many correction methods [56, 57, 58, 59, 60, 61] have been developed and have shown improvements, they have not succeeded in achieving high-order consistency and have even resulted in the loss of zero-order consistency.

### 1.2.2 SPH in PDEs-constrained optimization

As a mesh-free numerical method, SPH offers certain advantages in solving PDEs-constrained optimization problems. A brief review of general methods for addressing these problems, particularly in thermal conductivity optimization, is provided. In PDEs-constrained optimization, typical approaches for computing gradients, either explicitly or implicitly in simulation-based methods, include the adjoint technique [62, 63], automatic differentiation (AD) [64], and artificial neural networks (ANNs) [65, 66]. In addition, the augmented Lagrangian method [67] and sequential quadratic programming method [68] have been proposed to address these problems from an all-at-once perspective.

In thermal conductivity optimization, the goal is to redistribute a fixed amount of high thermal conductivity material to cool a heat-generating volume within specified boundaries, aiming to minimize temperature [69]. Simulation-based methods that address this problem using indirect optimal principles have been proposed and adopted. These include temperature gradient field homogenization [70, 71, 72], entropy generation minimization (EGM) [73, 74, 75, 76], and entransy dissipation extremum (EDE) [72, 77, 78, 79, 80]. Additionally, direct optimization targets, such as achieving the lowest average temperature [81, 63] and minimizing hot spot temperature [66, 82], have recently been employed. Besides these iterative methods based on deterministic principles, stochastic approaches, including the bionic optimization (BO) method [83, 84], cellular optimization (CA) method [85, 86], simulated annealing (SA), and genetic algorithm (GA) methods [87], have also been explored to solve thermal conductivity optimization problems.

Although these methods can achieve optimal solutions, they often involve a large number of variables in the spatial discretization of the domain, reducing efficiency and limiting their suitability for large-scale, reliability-sensitive problems. Therefore, developing an efficient yet straightforward approach, particularly one that leverages direct optimization targets and the all-at-once concept, holds great promise for addressing thermal conductivity optimization problems. SPH offers certain advantages in enhancing optimization efficiency due to its localized interactions with neighboring particles.

### 1.2.3 SPH-based libraries and verification

The SPH method has been implemented in several open-source scientific libraries with various focuses, including SPHinXsys [30], GPUSPH [88], SPHysics [89], DualSPHysics [90], AQUAgpushp [91], GADGET-2 [92], and GIZMO [93]. As these libraries continue to be actively developed, it is essential to ensure that modifications to the source code, whether for bug fixes or feature additions, do not compromise previously validated accuracy and functionality. This approach supports reliable

software development practices in scientific computing and enhances the efficiency of SPH method development and its applications.

To address these issues, implementing comprehensive testing, including unit tests, integration tests, regression tests, and system tests, provides concrete validation and verification procedures. Although these testing methodologies are standard in IT software development, applying them to scientific computing libraries, especially open-source ones, poses unique challenges due to the difficulty of establishing reliable test oracles. Among these testing methods, regression tests are crucial because they ensure that new changes do not introduce unexpected behavior or reduce result accuracy, thereby validating the output of scientific computing libraries under development. However, implementing regression tests within scientific computing libraries, particularly for meshless methods like SPH, presents significant challenges. Unlike mesh-based methods, which rely on fixed topologies, the Lagrangian nature of SPH, with its dynamic and variable particle distributions, complicates the establishment of standardized testing environments. Consequently, despite the growing popularity of meshless methods, there remains a noticeable lack of reliable testing environments specifically designed for them.

### 1.3 Aims and objectives

The fundamental objective of this thesis is to address the aforementioned issues of the SPH method to enhance its overall consistency and efficiency. This enhancement aims to establish SPH as an effective numerical method for both simulation and optimization, supported by a general and efficient development environment.

Regarding the challenges of achieving high-order consistencies in conservative SPH formulations, we propose the reverse KGC (RKGC) formulation. This approach is conservative and can satisfy first-order consistency when particles are relaxed based on the KGC matrix. Extensive numerical tests demonstrate that the new formulation significantly improves the accuracy of the Lagrangian SPH method. In particular, it addresses the long-standing issue of excessive dissipation in SPH free-surface flow simulations. Furthermore, when particles are fully relaxed, the RKGC formulation enhances the accuracy of the Eulerian SPH method, even when the ratio between the smoothing length and particle spacing is substantially reduced. This work is detailed in Paper I [94]

- B. Zhang, N. Adams, XY. Hu, Towards high-order consistency and convergence of conservative SPH approximations. *Computer Methods in Applied Mechanics and Engineering*. **433**: 11748, 2025

which has been attached in Appendix A.1.

To address the computational challenges of simulation-based optimization, we introduce a target-driven, all-at-once approach for PDE-constrained optimization and derive a splitting SPH method for optimizing thermal conductivity distribution in heat conduction problems. The optimization process is divided into manageable steps. It begins with a targeting step to enforce the direct target, which may increase PDE residuals, followed by an evolution step to adjust the design variable and reduce these residuals, and conducts with a PDE solution step to further advance the system before the next iteration. Unlike simulation-based methods, this approach bypasses the adjoint state equation and the converged state variable field, greatly simplifying and accelerating the optimization. Additionally, the use of an implicit

SPH splitting operator and a general regularization formulation improves information propagation and numerical stability. Typical examples of heat conduction optimization demonstrate that the current method yields optimal results comparable to those of previous methods while exhibiting considerable computational efficiency. This work is detailed in Paper II [95]

- B. Zhang, C. Zhang, XY. Hu, Target-driven splitting SPH optimization of thermal conductivity distribution. *International Journal of Heat and Mass Transfer*. **227**: 125476, 2024

which is attached in Appendix A.2.

To ensure the efficient development of SPH-based open-source scientific computing libraries, such as SPHinXsys, which forms the foundation of the current work, this paper introduces a method for establishing and implementing an automatic regression test environment, using SPHinXsys as an example. A reference database is first generated for each benchmark test based on multiple executions, capturing the variation range of key metrics through time-averaged, ensemble-averaged, and dynamic time warping methods. This approach accounts for uncertainties arising from parallel computing, particle relaxation, and physical instabilities. New results from source code modifications are then evaluated against this reference through curve similarity comparisons. Whenever the source code is updated, the regression test is automatically executed for all test cases, providing a comprehensive assessment of the validity of the current results. This regression test environment has been successfully implemented in all dynamic test cases within SPHinXsys, ensuring the validity of daily updates to the source code. This work is detailed in Paper III [96]

- B. Zhang, C. Zhang, XY. Hu, Automated regression test method for scientific computing libraries: Illustration with SPHinXsys. *Journal of Hydrodynamics*. **36(3)**: 466–478, 2024

which is attached in Appendix A.3.

## 1.4 Outline

The remainder of this thesis is structured as follows: Chapter 2 introduces the theory and fundamentals of the SPH method, the SPH discretization of the conservation equations for the Lagrangian and Eulerian SPH methods, as well as for the heat conduction equation, the consistency correction and its variational consistency analysis, particle relaxation, and transport-velocity formulation, as well as the splitting SPH method for optimization. Additionally, methods for regression testing of the SPH-based open-source scientific library are also discussed. Particularly, as detailed in Chapter 3, an RKGC conservative SPH formulation based on KGC-based particle relaxation is proposed to improve numerical consistency. A target-driven splitting SPH method is introduced for PDE-constrained optimization to greatly enhance optimization efficiency, specifically for thermal conductivity optimization in the current work. An automated regression test method is presented for the SPH-based open-source scientific library to support consistency and efficiency development. Finally, Chapter 4 concludes the key points of the thesis and suggests directions for future research.

## Chapter 2

# Methodology

This chapter provides a concise overview of the governing equations for fluid dynamics and heat conduction, along with their SPH discretizations. The theory and fundamentals of the SPH method are presented, including a discussion on consistency correction and its analysis. Following this, the splitting-SPH method designed for thermal conductivity optimization is introduced. Finally, the chapter briefly discusses the general regression testing methodology applicable to open-source scientific computing libraries, with a focus on SPH-based frameworks.

### 2.1 Governing equations

The governing equations in the Lagrangian framework for viscous flows consist of the mass and momentum conservation equations, written as

$$\begin{cases} \frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \\ \frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \end{cases}, \quad (2.1)$$

where  $\rho$  represents density,  $\mathbf{v}$  velocity,  $p$  pressure,  $\nu$  kinematic viscosity,  $\mathbf{g}$  gravity and

$$\frac{d(\bullet)}{dt} = \frac{\partial(\bullet)}{\partial t} + \mathbf{v} \cdot \nabla(\bullet) \quad (2.2)$$

refers to the material derivative.

The conservation equations for weakly compressible flows in the Eulerian framework are expressed as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (2.3)$$

where  $\mathbf{U} = (\rho, \rho \mathbf{v})$  denotes the vector of conserved variables, and  $\mathbf{F}(\mathbf{U})$  represents the corresponding fluxes.

For heat conduction, the steady-state temperature field within the thermal domain can be determined by solving the transient heat conduction governing equation given by

$$\frac{dT}{dt} = \nabla \cdot (k \nabla T) + \dot{Q}, \quad (2.4)$$

when the left-hand side (LHS) converges to zero. Here,  $T$  represents temperature,  $k$  thermal conductivity,  $C$  heat capacity, and  $\dot{Q}$  is the volume rate of the internal heat source. In addition, the Dirichlet boundary condition is given as

$$T = T_b, \quad (2.5)$$

where the surface temperature  $T_b$  is held constant. The Neumann boundary condition, which maintains a fixed heat flux rate, can be expressed as,

$$-k\nabla T \cdot \mathbf{n} = q_b, \quad (2.6)$$

where  $\mathbf{n}$  indicates the surface normal vector pointing outward.

## 2.2 Theory and fundamentals of the SPH method

In this section, the theory and fundamentals of the SPH method are summarized, followed by a discussion of SPH consistency correction and its variational analysis.

### 2.2.1 Integral interpolation and particle approximation

In SPH, to address the issue that the Dirac delta function cannot be used to establish a discrete model in integral interpolation, Lucy [21] and Gingold and Monaghan [97] introduced a smoothing kernel function to replace the Dirac delta function and approximate any smoothed field  $\psi(\mathbf{r})$  as

$$\psi(\mathbf{r}) = \int_{\Omega} \psi(\mathbf{r}^*) W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*, \quad (2.7)$$

where  $\mathbf{r}$  is the spatial coordinate, and  $W(\mathbf{r}, h)$  is the kernel function scaled by the smoothing length  $h$ , which determines the effective width of the smoothing kernel.

When the computational domain is discretized into a set of  $N$  particles as shown in Fig. 2.1, the integral in Eq. (2.7) is replaced by a summation over the values of

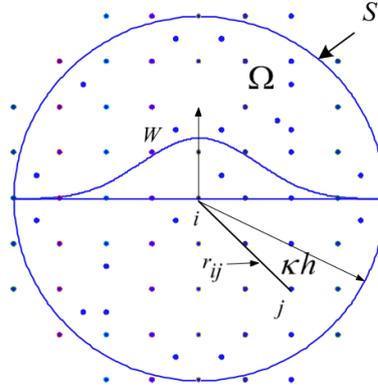


FIGURE 2.1: Sketch of particle approximation in the SPH method. Here,  $kh$  presents the support length of the smoothing kernel  $W$ .

adjacent particles in the support domain and is expressed as

$$\psi(\mathbf{r}_i) = \sum_{j=1}^N \psi(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h) dV_j. \quad (2.8)$$

Here, the indexes  $i$  and  $j$  denote the particle label, and the particle has mass  $m$  and density  $\rho$  at position  $\mathbf{r}$ , with  $dV_j$  denoting the differential volume element around the

particle. If we express the differential volume element  $dV_j$  as  $\frac{m_j}{\rho_j}$ , then the approximation of  $\psi(\mathbf{r}_i)$  can be written as

$$\psi(\mathbf{r}_i) = \sum_{i=1}^N \frac{m_i}{\rho_i} \psi(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h). \quad (2.9)$$

This summation is over particles lying within a circle of radius  $kh$  centered at particle  $i$  with position  $\mathbf{r}_i$  as shown in Fig. 2.1.

Based on the SPH kernel approximation expressed in Eq. (2.7), the gradient of a smooth field  $\psi(\mathbf{r})$  can also be approximated as

$$\nabla \psi(\mathbf{r}) = \int_{\Omega} \nabla \psi(\mathbf{r}^*) W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*. \quad (2.10)$$

By employing the integration by parts and applying Gauss's theorem, Eq. (2.10) can be derived as

$$\nabla \psi(\mathbf{r}) = - \int_{\Omega} \psi(\mathbf{r}^*) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*. \quad (2.11)$$

Applying particle approximation, described by Eqs. (2.7-2.9), Eq. (2.11) can be rewritten in SPH form as

$$\nabla \psi(\mathbf{r}_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} \psi(\mathbf{r}_j) \nabla W(\mathbf{r}_i - \mathbf{r}_j, h). \quad (2.12)$$

where  $\nabla W(\mathbf{r}_i - \mathbf{r}_j, h) = \nabla W_{ij} = \frac{\mathbf{e}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}}$  with  $\mathbf{e}_{ij} = \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|}$ .

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

$$\nabla \cdot \mathbf{F}(\mathbf{r}) = \int_{\Omega} [\nabla \cdot \mathbf{F}(\mathbf{r}^*)] W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{F}(\mathbf{r}_j) \nabla W_{ij}.$$

### 2.2.2 Consistency correction and variational analysis

In SPH discretization of the partial differential equation of fluid dynamics, the kernel approximation for the gradient operator acting on a smooth field  $\psi(\mathbf{r})$  can be expressed through a two-stage approach

$$\nabla \psi(\mathbf{r}) \approx \int_{\Omega} \nabla \psi(\mathbf{r}^*) W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = - \int_{\Omega} \psi(\mathbf{r}^*) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*. \quad (2.13)$$

While the first stage introduces smoothing errors by the kernel function, the second stage entails integration by parts, assuming the kernel function vanishes at the boundary of a compact support. Through Taylor expansion, for Eq. (2.13) one can find that the zero-order consistency condition is

$$\int_{\Omega} \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = 0, \quad (2.14)$$

and the first-order consistency condition is

$$- \int_{\Omega} (\mathbf{r}^* - \mathbf{r}) \otimes \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = \mathbf{I}, \quad (2.15)$$

where  $\mathbf{I}$  represents the identity matrix. Zero-order consistency condition allows rewriting the kernel approximation in two equivalent forms:

$$\begin{aligned}\nabla\psi(\mathbf{r}) &= \int_{\Omega} (\psi(\mathbf{r}) - \psi(\mathbf{r}^*)) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* \\ &\equiv - \int_{\Omega} (\psi(\mathbf{r}) + \psi(\mathbf{r}^*)) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*.\end{aligned}\quad (2.16)$$

By introducing particle summation, the first approximation in Eq. (2.16) can be further approximated for an SPH particle  $i$  as

$$\nabla\psi_i = \sum_j \psi_{ij} \nabla W_{ij} V_j, \quad (2.17)$$

where  $V_j$  is the volume of the neighbor particles within the support, and the particle-pair difference is  $\psi_{ij} = \psi_i - \psi_j$ . This form is often referred to as a symmetric or non-conservative form. Similarly, the second approximation in Eq. (2.16) can be also approximated as

$$\nabla\psi_i = - \sum_j (\psi_i + \psi_j) \nabla W_{ij} V_j, \quad (2.18)$$

where the particle-pair sum is employed. This form, known as the anti-symmetric or conservative form, ensures discrete conservation and is commonly chosen in classical SPH methods for discretizing physical conservation laws.

For the non-conservative form of Eq. (2.17), zero-order consistency is automatically satisfied as the particle-pair difference is used. To achieve first-order consistency, one requires that the approximation of Eq. (2.15) satisfies

$$- \sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j = \mathbf{I}. \quad (2.19)$$

To precisely fulfill the above condition, the KGC approach [51], introducing a correction matrix  $\mathbf{B}_i$  to adjust the gradient of the kernel function, can be employed, so that one has

$$- \sum_j \mathbf{r}_{ij} \otimes \mathbf{B}_i \nabla W_{ij} V_j = \mathbf{I}, \quad \mathbf{B}_i = \left( - \sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j \right)^{-1}. \quad (2.20)$$

With the KGC, Eq. (2.17) is modified as

$$\nabla\psi_i = \sum_j \psi_{ij} \mathbf{B}_i \nabla W_{ij} V_j. \quad (2.21)$$

Note that, introducing  $\mathbf{B}_i$  does not affect the zero-order consistency of Eq. (2.21). Also note that, although the non-conservation form is not desirable for the discretization of physical conservation laws, Eq. (2.21) is often used when the conservation is not a primary concern because it can reproduce the linear gradient and achieve second-order accuracy.

In the conservative form of Eq. (2.18), where the particle-pair sum other than the difference is used, the zero-order consistency condition becomes nontrivial as

$$\sum_j \nabla W_{ij} V_j = 0. \quad (2.22)$$

Litvinov et al. [47] proposed a particle relaxation process driven by a constant background pressure assuming invariant particle volume. After the particles are settled down or fully relaxed, Eq. (2.22) is satisfied for the zero-order consistency. To achieve first-order consistency, as a straightforward extension for Eq. (2.18), one may suggest

$$\nabla\psi_i = -\sum_j (\psi_i \mathbf{B}'_i + \psi_j \mathbf{B}'_j) \nabla W_{ij} V_j, \quad (2.23)$$

where  $\mathbf{B}'_i$  and  $\mathbf{B}'_j$  are some correction matrices for particles  $i$  and  $j$ , is able to reproduce a linear gradient similar to Eq. (2.21). How to obtain these correction matrices is not straightforward, and various attempts based on the original KGC matrix for the non-conservative form have been carried out. One widely used formulation, introduced by Oger et al. [50], is expressed as,

$$\nabla\psi_i = -\sum_j (\psi_i \mathbf{B}_i + \psi_j \mathbf{B}_j) \nabla W_{ij} V_j, \quad (2.24)$$

where the KGC matrix is applied for each particle pair separately.

Although it has been shown that the straightforward application of the KGC matrix is able to obtain improved results for some problems [56, 98, 99, 100] compared to the original conservative formulation in Eq. (2.18), it does not achieve zero- or first-order consistency for the conservative formulation. Specifically, Eq. (2.24) can be rewritten as

$$\nabla\psi_i = -\psi_i \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j + \sum_j \psi_{ij} \mathbf{B}_j \nabla W_{ij} V_j. \quad (2.25)$$

The first term on the right-hand side (RHS), mimicking Eq. (2.22), represents the zero-order consistency condition for incorporating the KGC, and the second term, again mimicking Eq. (2.21), represents the first-order consistency. However, the first term generally does not vanish even after particle relaxation driven by constant background pressure [47] due to the modification by the KGC matrix. In addition, the second term is different from the original form as the KGC matrix of neighboring particles is employed. Consequently, it does not guarantee first-order consistency either. These same issues also arise in other corrected formulations [59, 60, 61].

Therefore, we modify Eq. (2.25) by using the KGC matrix of particle  $i$  as

$$\nabla\psi_i = -\psi_i \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j + \sum_j \psi_{ij} \mathbf{B}_i \nabla W_{ij} V_j, \quad (2.26)$$

so that the second term is the same as Eq. (2.21) and achieves first-order consistency. If the first term also vanishes for achieving zero-order consistency, the entire formulation satisfies both consistencies at the same time. Motivated by employing constant background pressure for particle relaxation, we can consider  $\mathbf{B}_i$  and  $\mathbf{B}_j$  as “geometric stresses” dependent on the particle locations and use them to drive particle relaxation. After the particles are settled down or fully relaxed under such a KGC-driven particle relaxation, the first term vanishes. Note that Eq. (2.26) can be cast as the following anti-symmetric form

$$\nabla\psi_i = -\sum_j (\psi_i \mathbf{B}_j + \psi_j \mathbf{B}_i) \nabla W_{ij} V_j. \quad (2.27)$$

Comparing Eq. (2.27) with Eq. (2.24), one can find that the only difference is that, in the new formulation enabling both zero- and first-order consistencies, the KGC

matrix is employed in reverse order with respect to particles  $i$  and  $j$ . Therefore, Eq. (2.27) is denoted as a reverse KGC (RKGC) formulation.

The Lagrangian variational principle and other similar variational principles have been employed to analyze the conservation properties of the fluid in SPH method [57, 101, 102], and the principle of virtual work (PVW) has been adopted for the in-depth analysis of the free-surface simulation under the weakly compressible hypothesis [103]. Follow the PVW analyzing in NKGC [103] and SKGC [100] to analyze RKGC, where the work done by internal forces in an equilibrium system equals the work done by external forces due to the virtual displacement field  $\delta\mathbf{r}$ . By disregarding the work caused by the motion of solid boundaries and assuming zero pressure along the free-surface boundary [100], the PVW for the entire system can be expressed as the conservative form

$$\iiint_{\Omega} \nabla \cdot (p\delta\mathbf{r}) dV = 0, \quad (2.28)$$

and the theoretically equivalent non-conservative form

$$\iiint_{\Omega} \nabla p \cdot \delta\mathbf{r} dV = \iiint_{\Omega} -p\nabla \cdot \delta\mathbf{r} dV, \quad (2.29)$$

both indicating that the energy variation induced by the virtual displacement equals zero for the entire system, ensuring energy, linear- and angular-momentum conservations [57, 104].

By introducing particle approximation, the discrete form of Eq. (2.28) can be expressed as the summation of all discrete particles, we obtain

$$\sum_i \nabla \cdot (p\delta\mathbf{r})_i V_i = 0. \quad (2.30)$$

Since the RKGC formulation fulfills first-order consistency, the divergence operator  $\nabla \cdot (p\delta\mathbf{r})$  in Eq. (2.30) can be evaluated at each particle  $i$  as

$$\nabla \cdot (p\delta\mathbf{r})_i = \sum_j \left[ (p\delta\mathbf{r})_i \mathbf{B}_j + (p\delta\mathbf{r})_j \mathbf{B}_i \right] \nabla_i W_{ij} V_j, \quad (2.31)$$

with second-order accuracy. By substituting Eq. (2.31) into the left-hand side (LHS) of Eq. (2.30) and exploiting the anti-symmetric form of the present conservative discretization, one has

$$\begin{aligned} \sum_i \nabla \cdot (p\delta\mathbf{r})_i V_i &= \sum_i V_i \sum_j \left[ (p\delta\mathbf{r})_i \mathbf{B}_j + (p\delta\mathbf{r})_j \mathbf{B}_i \right] \nabla_i W_{ij} V_j \\ &= \sum_i \sum_j (p\delta\mathbf{r})_i \mathbf{B}_j \nabla_i W_{ij} V_i V_j \\ &\quad + \sum_i \sum_j (p\delta\mathbf{r})_j \mathbf{B}_i \nabla_i W_{ij} V_i V_j = 0, \end{aligned} \quad (2.32)$$

indicating the discrete variational consistency of the RKGC formulation. Actually, the conservative formulations, such as the standard and many other formulations, including SKGC featuring the anti-symmetric form, are able to maintain the discrete variational consistency of Eq. (2.30).

## 2.3 SPH discretization

In this section, the SPH method will be applied to hydrodynamics and heat transfer, and the corresponding discretized forms of the governing equations for fluid flow and heat conduction will be derived.

### 2.3.1 Weakly compressible SPH (WCSPH)

For WCSPH, an artificial equation of state (EOS) for weakly compressible flows is used to close Eq. (2.1) as

$$p = c_0^2 (\rho - \rho_0). \quad (2.33)$$

Here,  $\rho_0$  is the initial density, and  $c_0$  denotes the artificial sound speed. Setting  $c_0 = 10U_{max}$ , where  $U_{max}$  represents the anticipated maximum fluid speed, fulfills the weakly compressible assumption where the density variation remains around 1%.

The Riemann-SPH method [38] is employed here to discretize Eq. (2.1). Subsequently, the continuity and momentum equations are approximated as

$$\begin{cases} \frac{d\rho_i}{dt} = 2\rho_i \sum_j (\mathbf{v}_i - \mathbf{v}^*) \cdot \nabla W_{ij} V_j \\ \frac{d\mathbf{v}_i}{dt} = -\frac{2}{m_i} \sum_j P^* \nabla W_{ij} V_i V_j + 2 \sum_j \frac{v}{\rho_i} \frac{\mathbf{v}_{ij}}{r_{ij}} \frac{\partial W}{\partial r_{ij}} V_j + \mathbf{g}_i \end{cases}, \quad (2.34)$$

where  $\nabla W_{ij} = \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}$ , and  $\mathbf{e}_{ij} = \frac{\mathbf{r}_{ij}}{r_{ij}}$  is the unit vector. The particle-pair velocity  $\mathbf{v}^*$  and pressure  $P^*$ , respectively, are solutions obtained from the Riemann problem constructed along the interacting line of each pair of particles. Note that the particle-pair pressure  $P^*$  leads to an anti-symmetric form and hence momentum conservation. With a linearised Riemann solver, the solutions can be computed as

$$\begin{cases} \mathbf{v}^* = \bar{\mathbf{v}}_{ij} + (U^* - \bar{U}_{ij}) \mathbf{e}_{ij}, & U^* = \bar{U}_{ij} + \frac{1}{2} \frac{p_{ij}}{\rho_0 c_0} \\ P^* = \bar{p}_{ij} + \frac{1}{2} \beta \rho_0 c_0 U_{ij} \end{cases}. \quad (2.35)$$

Here,  $(\bar{\bullet})_{ij} = [(\bullet)_i + (\bullet)_j] / 2$  represents particle-pair average,  $\bar{U}_{ij} = -\bar{\mathbf{v}}_{ij} \cdot \mathbf{e}_{ij}$  and  $U_{ij} = -\mathbf{v}_{ij} \cdot \mathbf{e}_{ij}$ , represent the particle-pair average and difference of the particle velocity along the interaction line, respectively. The low-dissipation limiter is defined as  $\beta = \min(3 \max(U_{ij}/c_0, 0), 1)$ . Additionally, it should be noted that the particle-pair pressure  $P^*$  in Eq. (2.35) comprises two main components: a non-dissipative term denoted by  $\bar{p}_{ij}$ , and a dissipative term, which plays the role on satisfying the second law of thermodynamics for the overall WCSPH discretization, derived from the differences between particle pairs.

Similar to Refs. [104, 96] to avoid large density (and due to the invariant particle mass, the corresponding volume) errors in long-physical-time simulations, a density initialization scheme [105] is adopted to cancel the error accumulated in updating the density using a non-conservative formulation of the continuity equation Eq. (2.34). At the beginning of each new advection time step, the fluid density field of free-surface flows is reinitialized by

$$\rho_i = \max \left( \rho^*, \rho^0 \frac{\sum W_{ij}}{\sum W_{ij}^0} \right), \quad (2.36)$$

where  $\rho^*$  represents the density updated by Eq. (2.34) for the last several acoustic time steps and the superscript 0 denotes the initial reference value at the start of the simulation. For flows without the free surface, Eq. (2.36) simplifies to

$$\rho_i = \rho^0 \frac{\sum W_{ij}}{\sum W_{ij}^0} \quad (2.37)$$

which resets the density using the standard summation formulation [48].

### 2.3.2 Eulerian SPH (ESPH)

Following the methodology outlined in Refs. [106, 107], the Eulerian SPH discretization of Eq. (2.3) can be expressed in the same anti-symmetric or conservative form for both mass and momentum conservations as

$$\begin{cases} \frac{\partial}{\partial t} (\rho_i V_i) + 2 \sum_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} V_i V_j = 0 \\ \frac{\partial}{\partial t} (\rho_i \mathbf{v}_i V_i) + 2 \sum_j [(\rho \mathbf{v} \otimes \mathbf{v}) + p \mathbf{I}]_{E,ij}^* \cdot \nabla W_{ij} V_i V_j = 0 \end{cases} \quad (2.38)$$

Here, terms  $(\bullet)_{E,ij}^*$  denote numerical fluxes for each particle pair, determined by solutions of the Riemann problem [106]. The HLL Riemann solver [108, 109] incorporating a low-dissipation limiter [38] is adopted here to solve the Riemann problem. The solutions as numerical fluxes can be written as

$$\mathbf{F}^* = \frac{1}{2} \bar{\mathbf{F}}_{ij} + \beta \left( \frac{1}{2} \frac{S_R + S_L}{S_R - S_L} \mathbf{F}_{ij} + \frac{S_R S_L}{S_R - S_L} \mathbf{U}_{ij} \right). \quad (2.39)$$

under the assumption of  $S_L \leq 0 \leq S_R$  and  $S_l \neq S_R$ , which is validated in the low Mach number regime. Here,  $S_L$  and  $S_R$  represent the wave speeds estimated in the left and right regions of the Riemann problem, respectively, and are determined as

$$\begin{cases} S_L = \min \{u_L - c_L, u_R - c_R\} \\ S_R = \max \{u_L + c_L, u_R + c_R\} \end{cases} \quad (2.40)$$

The  $(\mathbf{U}_L, \mathbf{F}_L)$  and  $(\mathbf{U}_R, \mathbf{F}_R)$  denote conserved variables and fluxes in the left and right regions, while  $\mathbf{F}^*$  signifies the numerical flux determined in the intermediate region. In addition,  $\beta = \min(3 \max((U_L - U_R), 0), \bar{c})$ , where  $\bar{c} = (\rho_L c_L + \rho_R c_R) / (\rho_L + \rho_R)$ , represents the low-dissipation limiter introduced to handle intermediate states.

### 2.3.3 Heat conduction

The heat conduction governing equation in Eq. (2.4) can be discretized at each SPH particle  $i$  located at  $r_i$  with its neighboring particles  $j$  as follows,

$$\frac{dT_i}{dt} = 2 \sum_j \bar{k}_{ij} \frac{T_{ij}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i. \quad (2.41)$$

Here,  $T_{ij} = T_i - T_j$  indicates the inter-particle temperature difference, and  $V_j$  denotes the volume of neighboring particles  $j$ . The quantity  $\bar{k}_{ij} = (k_i + k_j)/2$  is the inter-particle average thermal conductivity.

Near the domain boundary, several layers of dummy particles are introduced to enforce different boundary conditions. Implementing the Dirichlet boundary condition is straightforward and involves imposing the temperatures

$$T_w = 2T_b - T_i, \quad (2.42)$$

at dummy particles implied by the wall boundary condition [110]. To implement the Neumann boundary condition, the discretization of Eq. (2.4) is modified into

$$\frac{dT}{dt} = \nabla \cdot (k\nabla T) + \dot{Q} + \dot{Q}^{\Gamma q}, \quad (2.43)$$

following the Ref. [111], where the heat flux in Eq. (2.6) is replaced by a volumetric term  $\dot{Q}^{\Gamma q}$ , which can be discretized as

$$\dot{Q}_i^{\Gamma q} = -q_b \sum_{j \in \Omega_j} (\mathbf{n}_i + \mathbf{n}_j) \cdot \nabla_i W_{ij} V_j. \quad (2.44)$$

Here,  $\Omega_j$  represents the boundary domain defined by the dummy particles. The unit vectors  $\mathbf{n}_i$  and  $\mathbf{n}_j$  are normal to the boundary evaluated at the respective positions of particle  $i$  and  $j$ .

## 2.4 Zero-order consistency satisfaction

In this section, the methods to satisfy zero-order consistency in the Lagrangian and Eulerian SPH methods are introduced.

### 2.4.1 KGC-based particle relaxation

The original relaxation method (denoted as P relaxation or PR), as detailed in Ref. [47, 112], operates by executing particle relaxation driven by a constant background pressure to achieve the zero-order consistency condition as outlined in Eq. (2.22). Initially, the particles are arranged in a lattice with the spacing of  $\Delta x$ . Their positions are then randomly shifted with a small distance. Subsequently, the method iteratively adjusts particle positions to rectify zero-order integration errors, with the correction at each relaxation step determined by

$$\Delta \mathbf{r}_i = \alpha (\Delta x)^2 \sum_j p^0 \nabla W_{ij} V_j. \quad (2.45)$$

Here,  $p^0$  the constant background pressure, set to the unit for simplicity. The parameter  $\alpha = 0.2$  is chosen as the limitation to avoid excessive time-step size to ensure numerical stability until the consistency error reaches a sufficiently small value. The selection of the value of  $\alpha$  follows the general selection of CFL number in the SPH method, where 0.25 is adopted [48], and we opt for a slightly smaller value to enhance stability. During the relaxation process, the particle volume is invariant, so the summation of particle volume is equal to the initial volume. Additionally, the relaxed particle distribution exhibits the partition of unity property [47], meaning there is neither a gap between nor an overlap of the volumes defined by particles.

To incorporate the proposed RKGC formulation, we introduce the KGC-based relaxation (denoted as B relaxation or BR), where the particle relaxation is driven by the “geometric stress” or the KGC matrix to achieve zero-order consistency condition as suggested by the first term in Eq. (2.26). Similarly to P relaxation, the iterative

correction on particle positions at each relaxation step is modified as

$$\Delta \mathbf{r}_i = \alpha (\Delta x)^2 \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j. \quad (2.46)$$

Note that the KGC matrix for each particle is recomputed by Eq. (2.20) before each iteration step. Similar to P relaxation, B relaxation also results in a uniformly distributed particle distribution, with body-fitted particles for complex geometries. In addition, the B relaxation process also does not change the total volume, and the relaxed particle distribution still maintains the underlying physics for the simulation. The KGC-based particle relaxation is used for generating the initial particle distribution for Eulerian SPH to combine with the KGC formulation. The general threshold to stop the relaxation is when the maximum zero-order consistency error is smaller than  $10^{-5}$ , where the average error is generally one-tenth of the maximum error.

## 2.4.2 KGC-based transport-velocity formulation

In the Lagrangian SPH method, the transport-velocity formulation introduced by Adami et al. [113] is employed to enhance, rather than precisely achieve, zero-order consistency, and to prevent particle clustering under conditions of negative pressure. The method involves a single correction step during each time step, where the transport-velocity, denoted by  $\tilde{\mathbf{v}}$ , governs the updates of particle positions from one step to the next according to

$$\frac{d\mathbf{r}_i}{dt} = \tilde{\mathbf{v}}_i, \quad (2.47)$$

and the generation of transport-velocity is associated with the acceleration induced by the repulsive force or the residual force, as indicated in Eq. (2.45). This residual force also leads to a self-relaxation mechanism that regularizes the particle distribution, allowing it to approximate a configuration with low consistency error [47]. The KGC-based transport-velocity formulation is proposed similarly, except that the residual force for the single correction step is associated with “geometry stress” in Eq. (2.46). Note that both transport-velocity formulations only slightly modify the positions of the particles without modifying the velocity or the momentum of the entire system. KGC-based transport velocity has been implemented in the Lagrangian SPH method for internal flow, while it is not employed for the free-surface flow following the general practices of SPH simulations [50, 98, 100].

## 2.5 Splitting SPH optimization method

### 2.5.1 Implicit scheme

It is well known that traditional implicit schemes often require large-scale matrix inversion or iterations across the entire system, which can lead to significant memory demands and challenges in parallelization. To overcome these challenges, we employ a splitting operator based implicit scheme to advance Eq. (2.41). The implicit solving step is divided into individual particle-by-particle operations, and each evolves a small system that is easy to inverse. One commonly used approach for this purpose is the second-order Strang splitting technique [114], shown as

$$S_i^{(\Delta t)} = D_1^{(\frac{\Delta t}{2})} \circ D_2^{(\frac{\Delta t}{2})} \circ \dots \circ D_i^{(\frac{\Delta t}{2})} \dots \circ D_{N_i-1}^{(\frac{\Delta t}{2})} \circ D_{N_i}^{(\frac{\Delta t}{2})} \circ D_{N_i}^{(\frac{\Delta t}{2})} \circ D_{N_i-1}^{(\frac{\Delta t}{2})} \circ \dots \circ D_i^{(\frac{\Delta t}{2})} \dots \circ D_2^{(\frac{\Delta t}{2})} \circ D_1^{(\frac{\Delta t}{2})}. \quad (2.48)$$

Here, the operator  $S_i^{(\Delta t)}$  represents the complete step for advancing the equation.  $N_i$  refers to the total number of particles, and  $D_i$  represents the splitting operator corresponding to particle  $i$ . The update of the variable for the entire field involves a forward sweep of all particles for half a time step, followed by a backward sweep for another half time step [115].

Within the local implicit formulation, Eq. (2.41) can be rewritten as

$$\frac{dT_i}{dt} = 2 \sum_j \bar{k}_{ij} \frac{T_{ij}^{n+1}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i^{n+1}, \quad (2.49)$$

where  $T_{ij}^{n+1} = T_{ij}^n + dT_i - dT_j$ . The terms  $dT_i$  and  $dT_j$  represent the incremental changes for particle  $i$  and its neighboring particles  $j$  at each advancing time step. For brevity, we introduce the coefficient

$$B_j = 2 \bar{k}_{ij} \frac{1}{r_{ij}} \nabla_i W_{ij} V_j dt, \quad (2.50)$$

and the residual of Eq. (2.49), without considering the increment, has the form

$$E_i = - \sum_j B_j T_{ij}^n - \dot{Q}_i^{n+1} dt. \quad (2.51)$$

The implicit formulation of Eq. (2.49) can be further expressed as

$$E_i = \left( \sum_j B_j - 1 \right) dT_i - \sum_j B_j dT_j. \quad (2.52)$$

To determine the incremental changes for temperature, we employ the gradient descent method [116] by reducing the LHS of Eq. (2.51) following its gradient. The gradient  $\nabla E_i$  with respect to the variable  $(dT_i, dT_1, dT_2, \dots, dT_N)^T$ , where  $N$  gives total number of all neighboring particles, can be obtained as

$$\nabla E_i = \left( \sum_j B_j - 1, -B_1, -B_2, \dots, -B_N \right)^T. \quad (2.53)$$

We set

$$(dT_i, dT_1, dT_2, \dots, dT_N)^T = \eta_i \nabla E_i, \quad (2.54)$$

where  $\eta_i$  represents the learning rate [116] for the particle  $i$ . Substituting Eqs. (2.53) and (2.54) into Eq. (2.52), the learning rate can be obtained as

$$\eta_i = \left( \left( \sum_j B_j - 1 \right)^2 + \sum_j (B_j)^2 \right)^{-1} E_i. \quad (2.55)$$

According to Eqs. (2.53) and (2.54), the incremental change in temperature of particle  $i$  and all its neighbors can be obtained and updated as

$$\begin{cases} T_i^{n+1} = T_i^n - dT_i = T_i^n + \eta_i (\sum_j B_j - 1) \\ T_1^{n+1} = T_1^n - dT_1 = T_1^n - \eta_i B_1 \\ T_2^{n+1} = T_2^n - dT_2 = T_2^n - \eta_i B_2 \\ \dots \\ T_N^{n+1} = T_N^n - dT_N = T_N^n - \eta_i B_N \end{cases} \quad (2.56)$$

Note that, Eq. (2.56) involves updating the variables for particle  $i$  and its neighboring particles simultaneously. When a shared-memory parallelization is employed, conflict may arise when multiple threads attempt to update the values of a single particle pair simultaneously. To address this issue, we have implemented a splitting Cell Linked List method [117]. This method effectively prevents conflicts by ensuring that neighboring particles are located in the same cell or in adjacent cells that are distributed among the same threads. Also note that, for an explicit integration of the thermal diffusion equation, the maximum allowable time step can be defined as

$$\Delta t_d = 0.5 \frac{\rho C h^2}{k_{max}}. \quad (2.57)$$

Since the implicit scheme is employed here for obtaining the steady solution of the Eq. (2.41), the time step size is chosen as a large value of  $10\Delta t_d$  without considering the temporal accuracy.

## 2.5.2 Evolution of design variable

The residual  $e_i^*$  for particle  $i$  in the PDE is calculated as

$$e_i^* = \sum_j (k_i + k_j) \frac{T_{ij}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i. \quad (2.58)$$

Once the target is imposed on this particle, the PDE residual deviates from its original value and will be recovered by modifying the design variable  $k$  on particle  $i$  and its neighboring particles  $j$ . This process can be represented by the pseudo-time evolution of following equation

$$\frac{dk_i}{d\tau} = \sum_j (T_i^c - T_j) (k_i^{m+1} + k_j^{m+1}) \frac{1}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i + e_i^*. \quad (2.59)$$

Here,  $k_i^{m+1} + k_j^{m+1} = k_i^m + dk_i + k_j^m + dk_j$ , where  $m$  is the previous time step and  $dk_i$  and  $dk_j$  represent increments after the new time step. The implicit splitting operator introduced in Section 2.5.1 is utilized. Similar to the Eq. (2.54), a linear system is formed with respect to  $(dk_1, dk_2, \dots, dk_{N-1}, dk_N)^T$ . Note that, the pseudo-time derivative on the LHS is essential for the stable evolution of  $k$ . If this term is omitted, the diagonal entries of the matrix for the linear system become

$$\left( \eta_1 \sum_j^n B_{1j}, \eta_2 \sum_j^n B_{2j}, \dots, \eta_N \sum_j^n B_{Nj} \right)^T. \quad (2.60)$$

It is observed that the magnitudes of diagonal entries in the matrix can be significantly smaller than those of non-diagonal entries, potentially leading to numerical instability [118]. On the contrast, when the pseudo-time derivative term is included, the linear system transforms into

$$\left( \eta_1 \sum_j^n (B_{1j} - 1), \eta_2 \sum_j^n (B_{2j} - 1), \dots, \eta_N \sum_j^n (B_{Nj} - 1) \right)^T, \quad (2.61)$$

whose diagonal entries becomes dominant, and therefore stabilize the evolution of the design variables. In addition, since  $k$  is a material property and should be non-negative, it is clipped at a lower bound of 0.0001 during each iteration.

### 2.5.3 Numerical regularization

After the evolution of the design variable, it is necessary to apply numerical regularization, serving two essential purposes. One is that, as previously mentioned in the Ref. [63], the regularization plays a critical role in maintaining numerical stability and obtaining a smooth solution. Secondly, it helps prevent over-fitting and avoids finding trivial local optima only. In this study, we introduce a diffusion analogy approach for regularizing the distribution of the design variable, i.e. the thermal conductivity  $k$  is treated as the variable again in the pseudo-time SPH discretized diffusion equation, given as

$$\frac{dk_i}{d\tau} = 2 \sum_j \mu \frac{k_{ij}}{r_{ij}} \nabla_i W_{ij} V_j, \quad (2.62)$$

where  $k_{ij} = k_i - k_j$  and  $\mu$  is the artificial diffusion coefficient used to control the rate of regularization. We choose  $\mu$  to be general in the range  $1 \sim 2$  according to the target strength. The coefficient also undergoes a similar dynamical adjustment strategy as the target strength because smaller  $\beta$  usually require less regularization to achieve a smooth field. Note that the pseudo-time derivative term is also used in Eq. (2.62) to ensure that the diagonal is dominant.

## 2.6 Regression test method

### 2.6.1 Time-averaged strategy

In the time-averaged strategy, as the system consistently reaches a steady state through the relaxation process, metrics such as the time-averaged mean and variance are employed for comparison and testing purposes.

The generation of the reference database under this strategy involves updating the time-averaged mean and variance through multiple executions until their variations converge. For each update (e.g., the  $n$ th execution), the mean  $M^n$  and variance  $\sigma^n$  of the obtained result  $x$  from the current execution can be calculated as follows:

$$\begin{cases} M^n = \frac{1}{l-k} \sum_{i=k}^l x_i^n \\ \sigma^n = \frac{1}{l-k} \sum_{i=k}^l (x_i^n - M^n)^2 \end{cases}, \quad (2.63)$$

where  $i$  is the index of a data point,  $l$  is the total number of data points, and  $k$  is the index of the starting point of the steady state. The mean  $M^*$  and variance  $\sigma^*$  in the regression test metrics are then updated based on the results from the  $n$ th computations. Specifically,  $M^*$  is updated as

$$M^* = (M^n + M^* \times (n - 1)) / n. \quad (2.64)$$

Note that, instead of storing all previous means  $n$  times, the summation of the mean is recursively updated as a decaying average of all previous means for increased efficiency. Then,  $\sigma^*$  is updated as

$$\sigma^* = \max(\sigma^n, \sigma^*), \quad (2.65)$$

indicating that the variance is always updated to the maximum variation range. After the relative difference between the newly updated metrics and the previous ones is smaller than thresholds in several successive executions (usually 4 in practical applications), the  $M$  and  $\sigma$  are stored as the reference database. It is worth noting that the variation of the metrics in two successive runs being smaller than the threshold is only a hint of convergence. This should happen several times successively to ensure a real stable convergence. Therefore, once such variation is larger than the threshold, the count of the converged successive executions will be reset to zero.

For the regression test, the mean value and variance of the new result are compared with the metrics of the reference database. The correctness of the new result and the compatibility of the modified code with the previous version are determined based on the following conditions:

$$\begin{cases} |M - M^*| \leq \alpha M^* \\ \sigma \leq \sigma^* \end{cases}. \quad (2.66)$$

If these conditions are satisfied, the new result is considered correct, and the modified code is deemed compatible with the previous version. The parameter  $\alpha$  is chosen based on the specific type of dynamics problem. In solid dynamics,  $\alpha$  is set to 0.05, while for fluid dynamics, it is set to 0.1.

## 2.6.2 Ensemble-averaged strategy

In the ensemble-averaged strategy, the result curves obtained from simulation runs typically exhibit similarities within a certain variation range. This range is defined by the metrics of ensemble-averaged mean and variance. For the  $n$ th execution, the metrics for each data point  $i$  are updated based on the previous values and the new results. The ensemble-averaged mean  $M_i^*$  at a data point  $i$  is updated as:

$$M_i^* = (x_i^n + M_i^{n-1} \times (n - 1)) / n. \quad (2.67)$$

Here,  $x_i^n$  represents the newly obtained data point, and  $M_i^{n-1}$  is the previous mean. Similarly to the time-averaged strategy, the new variance  $\sigma_i^*$  is updated as

$$\sigma_i^* = \max(\sigma_i^*, \sigma_i^{n-1}, (0.01 * (M_{i,max}^* - M_{i,min}^*))^2), \quad (2.68)$$

where the last term is a secure value introduced to create a variation range and prevent a zero maximum variance for results from different computations. This secure

value is determined based on the maximum and minimum values of the local result. The convergence criteria are the same as those in the time-averaged strategy, requiring successive executions with sufficiently small variations of the updated mean and variance to terminate the metrics updating. With the metrics of the reference database available, the regression test following code modification is conducted for all data points using the following condition:

$$\sqrt{(x_i^n - M_i)^2} \leq \sigma_i. \quad (2.69)$$

If there is any data point that does not satisfy this condition, it indicates that the code modification needs to be reviewed and corrected.

### 2.6.3 Dynamic time warping (DTW) strategy

The maximum DTW distance is utilized as the regression test metric and is updated after each execution until its variation converges to a certain threshold. With the initial value for the first computation set as  $D_{0,0} = 0$ , the maximum distance for the  $n$ th execution is calculated as

$$D^* = \max(D^*, D_{0,n}, D_{1,n}, \dots, D_{n-2,n}, D_{n-1,n}), \quad (2.70)$$

where the subscript, e.g.,  $D_{n-2,n}$  denotes the distance between the  $(n-2)$ th and  $n$ th computational results. Similar to the other two strategies, after the variation of  $D^*$  converges to a given threshold in successive several executions, the  $D^*$  and several results (usually  $3 \sim 5$  in practical applications) with all data points are stored for the regression test. For the regression test, if the DTW distances between the new result after code modification and each result in the reference database satisfy the condition:

$$(D_1, D_2, \dots, D_k) \leq D^*, k = 3 \sim 5, \quad (2.71)$$

the new result is regarded as acceptable. Otherwise, if any distances exceeds the threshold  $D^*$ , it indicates unexpected behavior, and the code should be checked and corrected.

## 2.7 Summary

This chapter covers the fundamentals and theoretical aspects of the SPH method, beginning with integral interpolation and particle approximation. It discusses the traditional SPH discretization of governing equations for fluid dynamics and heat conduction. The consistency correction in SPH formulation and its variational analysis are also explained, addressing the method for the satisfaction of zero-order consistency. The splitting SPH method for optimizing thermal conductivity is summarized, and the implementation of general regression tests for SPH-based open-source libraries is briefly introduced.



## Chapter 3

# Summaries of publications

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

### 3.1 Towards high-order consistency and convergence of conservative SPH approximations

Bo Zhang, Nikolaus Adams, Xiangyu Hu

#### 3.1.1 Summary of the publication

In SPH, for a given kernel function, there are two typical particle approximation methods that approximate the integral in the kernel approximation by summing over all particles within the support domain. One method is the non-conservative, or symmetric, form, which uses the particle pair difference and is expressed in Eq. (2.17). This form naturally satisfies zero-order consistency, and higher-order consistencies can be achieved, although at the expense of computational efficiency, by using kernel gradient correction (KGC) [51] or other methods based on similar ideas [119, 120, 121, 122]. However, its applicability for discretizing conservation laws in practical simulations is limited due to a lack of discrete conservation. The other form is the conservative, or anti-symmetric, form, which uses the particle pair average and is expressed as Eq. (2.18). This form preserves the conservation property and is commonly used for discretizing momentum equations. While zero-order consistency can be achieved through particle relaxation, attempts to achieve higher-order consistency through straightforward extensions of the KGC, as proposed by many researchers [56, 57, 58, 99, 100, 98], have shown improved results for certain problems but have ultimately proven insufficient for satisfying zero- or higher-order consistencies.

In this paper, we propose the reverse KGC (RKGC) formulation, read as (2.27) and it can be rewritten as Eq. (2.26), where the first term can vanish with particle relaxation based on the KGC matrix, and the second term achieves first-order consistency, the same as the non-conservative form. After the particles settle or are fully relaxed through KGC-based particle relaxation, the RKGC formulation satisfies both zero- and first-order consistencies simultaneously. When the RKGC formulation is employed in the momentum conservation equation for the Lagrangian SPH method, as well as in both the mass and momentum conservation equations for the Eulerian SPH method, numerical examples such as the Taylor Green vortex flow and the lid-driven cavity, all demonstrate improved numerical accuracy. More importantly, RKGC formulation shows good energy conservation properties, resolving the long-standing high-dissipation issue for simulating free-surface flows in SPH. The pressure contour of the oscillating problem at two different instants is

presented in Fig. 3.1, indicating the robust free-surface profile and smooth pressure fields achieved by the RKG formulation. The time evolution of the decay of the

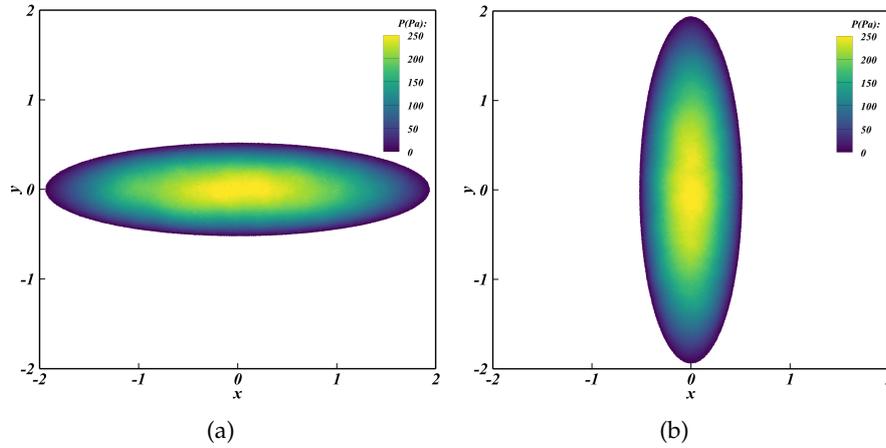


FIGURE 3.1: Oscillation drop: Snapshots of the free-surface profile and the pressure contour obtained by the RKG ( $\Delta x = 0.005$ ). (a)  $t=20.5s$ ; (b)  $t=22.9s$ .

normalized mechanical energy in the oscillating drop problem is shown in Fig. 3.2. It is noted that the RKG formulation preserves energy quite well, even at low reso-

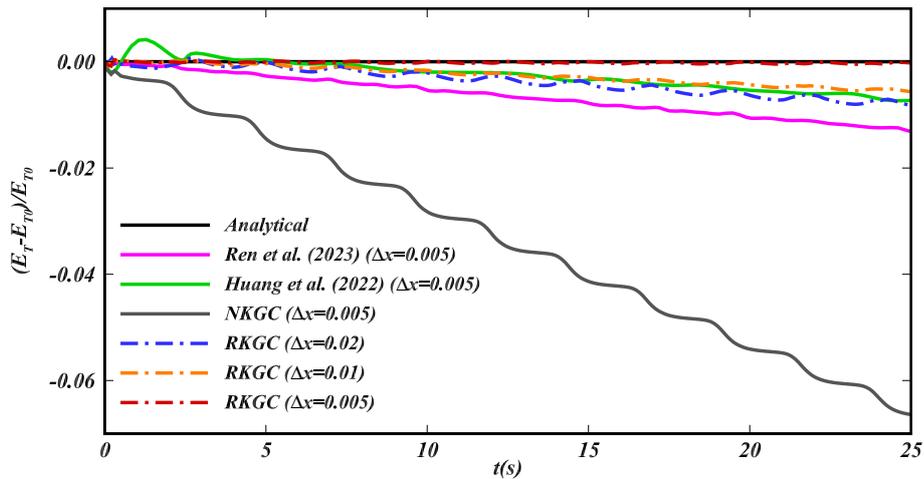


FIGURE 3.2: Oscillating drop: Time evolution of the decay of the normalized mechanical energy obtained by different formulations.

lutions, and the energy conservation properties improve with increasing resolution. Similarly, for other free-surface flows, such as standing waves, progressive wave propagation, dam-break flow, and three-dimensional oscillating wave surge converters (OWSC), the RKG formulation all demonstrates good energy conservation properties and improved numerical accuracy.

### 3.1.2 Individual contributions of the candidate

This article [123] was published in the international peer-reviewed journal *Computer Methods in Applied Mechanics and Engineering*. My contributions to this work include the development of the method and the corresponding computer code for its implementation. I conducted simulations, analyzed the results, and wrote the manuscript for publication.

## 3.2 Target-driven splitting SPH optimization of thermal conductivity distribution

Bo Zhang, Chi Zhang, Xiangyu Hu

### 3.2.1 Summary of the publication

As a general numerical method, SPH can be employed to solve optimization problems and offers potential advantages due to its interaction only related to the neighboring particles. In this study, a PDE-constrained optimization problem [11] is addressed using the target-driven splitting SPH approach, focusing on optimizing the thermal conductivity distribution for two-dimensional heat conduction problems. The thermal domain, as illustrated in Fig. 3.3, is subject to different boundary conditions and may include internal heat sources. Two typical boundary conditions,

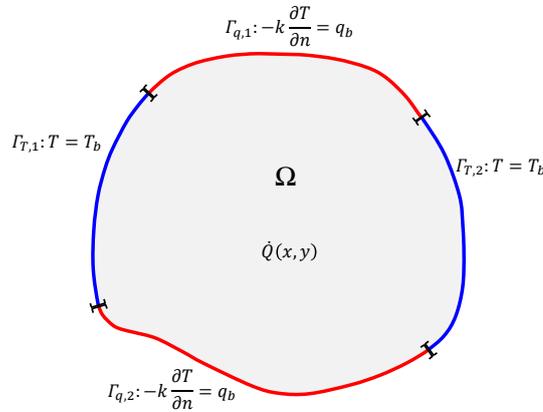


FIGURE 3.3: Illustration of the 2D heat conduction problem.  $\Omega$  denotes the thermal domain;  $\Gamma_T$  and  $\Gamma_q$  represent constant surface temperature and heat flux boundaries, respectively, and  $\dot{Q}$  indicates the presence of an internal heat source.

Dirichlet and Neumann, are considered. The goal of the optimization is to determine a thermal conductivity distribution that minimizes the average steady-state temperature while ensuring that the average thermal conductivity remains constant.

The target-driven method divides the optimization process into several manageable steps, with each step only being weakly coupled to the others. This allows both the design variable (thermal conductivity) and the state variable (temperature) to be treated as both optimization variables by introducing the residual recovery concept and achieving "all-at-once" optimization approaches [25, 26], improving the optimization efficiency. Specifically, a targeting step is used to incrementally impose the direct target, which may increase the PDE residuals. These residuals are then recovered through an evolution step of the design variables. Following this, a PDE solution step further reduces the residuals and prepares for the next iteration. In addition, the splitting SPH method, an implicit scheme based on a splitting operator, is employed to solve the temperature and thermal conductivity fields, advancing the PDE and the design variable evolution. Finally, a diffusion-analogy regularization approach is developed to ensure numerical stability and avoid local optima.

Typical heat conduction optimization examples have been conducted, demonstrating that the proposed method yields optimal results comparable to previous extensive simulation-based approaches [70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]. Furthermore, the method shows significant computational efficiency, requiring only a

few times of the computational cost of obtaining a steady-state solution to achieve the optimized results. For example, a square thermal domain measuring 1m on each side, with four Gaussian heat sources, as depicted in Fig. 3.4(a), is considered. Two heat sinks, each covering 20% of the side length, are symmetrically positioned at the center of opposite boundaries, maintaining a constant temperature. The steady

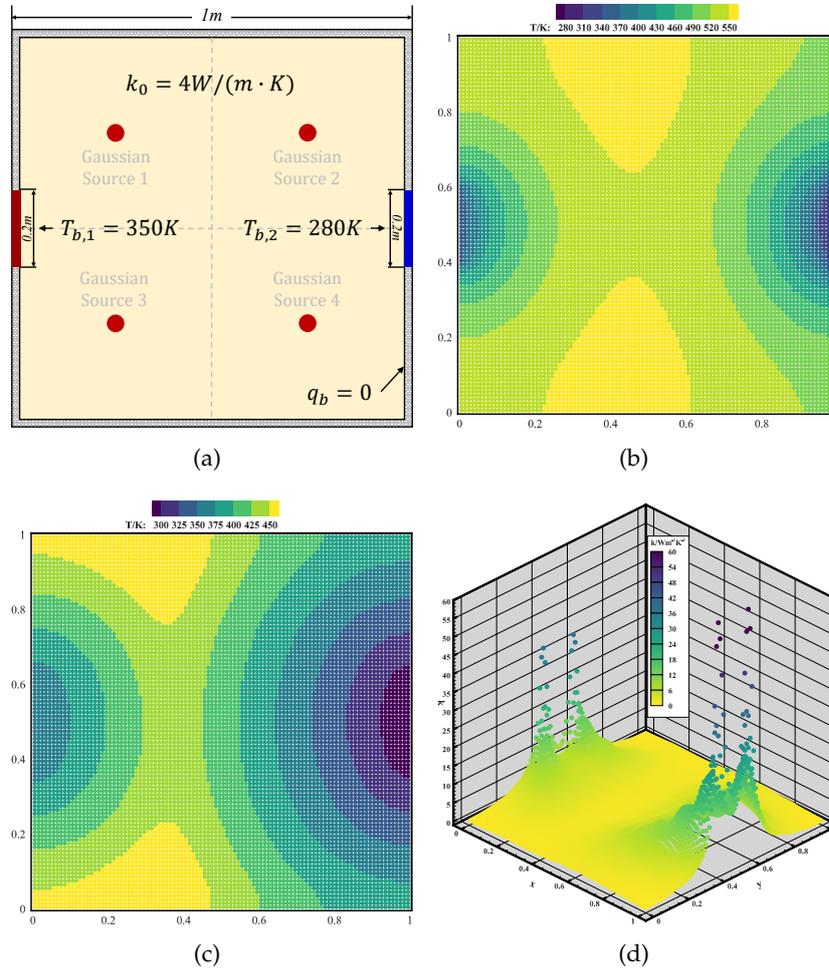


FIGURE 3.4: The 2/10 sinks with the Gaussian distributed heat source. (a) Problem setup; (b) Temperature distribution with uniform thermal conductivity; (c) Optimized temperature distribution; (d) Optimized thermal conductivity distribution.

temperature distribution with uniform thermal conductivity is shown in Fig. 3.4(b), while the optimized results using the current method are presented in Figs. 3.4(c) and 3.4(d). These results demonstrate more effective temperature reduction compared to the AA and TGH methods [63], with less than ten times the computational cost required to achieve a steady-state solution. Other examples, including different configurations of heat sinks and heat sources, similarly show improved efficiency.

### 3.2.2 Individual contributions of the candidate

This article [94] was published in the international peer-reviewed journal *International Journal of Heat and Mass Transfer*. My contributions to this work include the development of the method and the corresponding computer code for its implementation. I conducted simulations, analyzed the results, and wrote the manuscript for publication.

### 3.3 Automated regression test method for scientific computing libraries: Illustration with SPHinXsys

Bo Zhang, Chi Zhang, Xiangyu Hu

#### 3.3.1 Summary of the publication

As a fully Lagrangian meshless method, SPH has been extensively studied and used to simulate a wide range of scientific problems. SPHinXsys [30], an open-source multi-physics and multi-resolution scientific computing library based on SPH, provides an environment for developing SPH algorithms and engineering applications, including the work introduced in Secs. 3.1 and 3.2. The continuous development of these algorithms and their applications highlights the necessity for a regression testing environment [124] to ensure consistency throughout version releases and improve development efficiency.

In this paper, we propose a general method for establishing and implementing an automated regression testing environment for scientific computing libraries, using the open-source multi-physics library SPHinXsys as a case study. The underlying principle of the regression test involves comparing the similarity between verified curves, also referred to as time series results, generated from previously validated executions in the reference database and newly obtained results following code modifications. As illustrated in Fig. 3.5, a reference database for each bench-

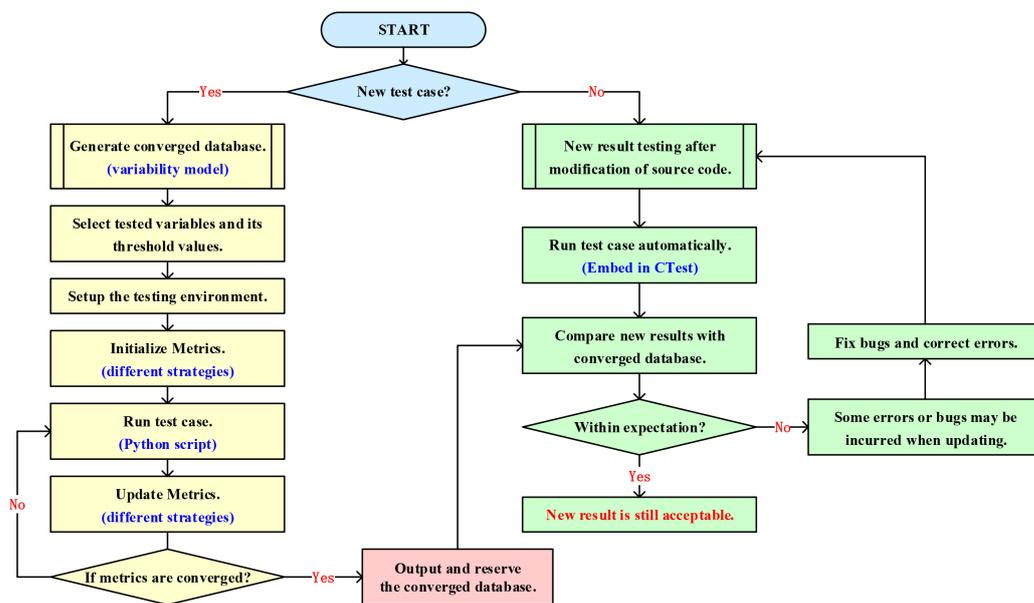


FIGURE 3.5: The flow chart for the regression test. The left part is for generating the reference database, and the right one is for testing the new result obtained after code modifications.

mark test is generated from observed data across multiple executions for newly added test cases. This database encapsulates the maximum variation range of metrics for different strategies, including time-averaged, ensemble-averaged, and dynamic time warping methods [125, 126] based on different curve classifications. For instance, in many fluid dynamics problems, data series fluctuate around a constant value after reaching a steady state, making the time-averaged strategy suitable. In

solid dynamics, data series exhibit similar variation patterns for each computation, so the ensemble-averaged strategy is selected. Data series that experience rapid, scattered variation patterns or large high-frequency fluctuations will utilize the dynamic time warp strategy. With the reference database established, new results obtained after source code modifications undergo testing through a curve-similarity comparison against this database.

This regression test environment has been successfully implemented across all dynamic test cases within SPHinXsys, covering fluid dynamics [38, 105], solid mechanics [127], fluid-solid interaction (FSI) [43], thermal and mass diffusion [128], reaction and diffusion [128], electromechanics [128], and their multi-physics couplings, demonstrating robust capabilities in testing various problems. Moreover, the underlying principles are generic and can be easily adapted for use with other libraries, achieving equal effectiveness.

### **3.3.2 Individual contributions of the candidate**

This article [129] was published in the international peer-reviewed journal *Journal of Hydrodynamics*. My contributions to this work include the development of the method and the corresponding computer code for its implementation. I conducted simulations, analyzed the results, and wrote the manuscript for publication.

## Chapter 4

# Discussion and outlook

The motivation behind this thesis was threefold: to advance the consistency and convergence of conservative SPH approximations for improved numerical accuracy, to enhance the efficiency of thermal conductivity optimization in heat transfer problems using a target-driven splitting SPH method, and to develop an automated regression testing framework for scientific computing libraries like SPHinXsys to ensure code reliability and accuracy. These objectives have been successfully achieved through the application of the proposed methods to various practical cases. In conclusion, this section provides a discussion of the current work and remarks on potential directions for future research.

### 4.1 Discussions

As a mesh-free method, SPH offers distinct advantages for modeling many engineering problems and has evolved into a general numerical approach for simulation and optimization, with advancements in consistency and efficiency. However, achieving high-order consistency in the conservative SPH formulation remains a significant challenge. While zero- and higher-order consistencies can be obtained using particle-pair differences and kernel gradient correction (KGC) approaches [51] for SPH gradient approximations, their applicability for discretizing conservation laws in practical simulations is limited by a lack of discrete conservation. The standard anti-symmetric SPH approximation achieves conservation and can satisfy or enhance zero-order consistency through particle relaxation based on a constant background pressure [47, 112] or the transport-velocity formulation [113, 96]. Nevertheless, high-order consistency in SPH remains necessary to ensure that truncation errors decay with increasing particle resolution. To address this, approaches have been developed to improve numerical accuracy and consistency by integrating KGC with anti-symmetric formulations [56, 57, 58] and/or particle relaxation [98]. These include methods such as using the average correction matrix [59, 60, 61] and implementing separate corrections for each particle pair [50, 99]. However, these methods still fail to achieve high-order consistency and often lose zero-order consistency, even with particle relaxation.

In Paper I, we introduce the reverse KGC (RKGC) formulation designed for conservative SPH approximations. In contrast to prior conservative KGC formulations, without correction (NKGC) in Eq. (2.18), with the original straightforward KGC (SKGC), RKGC is able to precisely fulfill both zero- and first-order consistencies of the gradient operator. The formulation incorporates a particle relaxation driven by the KGC matrix and is split into two parts: the first part addresses zero-order consistency and vanishes during particle relaxation, while the second part ensures first-order consistency and accurately reproduces linear gradients. The formulation notably improves numerical accuracy in Lagrangian SPH simulations, with evidence

from both internal and free-surface flow. In particular, it exhibits very good energy conservation properties and resolves the long-standing high-dissipation issue for the SPH simulation of free-surface flows. For example, in the standing wave problem, RKGC formulation is able to preserve the energy very well, suggesting very small numerical dissipation. However, NKGC exhibits rapid energy decay, even when the smoothing length is increased to  $h = 2.0\Delta x$ , as shown by Khayyer et al. [130], and the SKGC formulation even leads to an increase in the energy, as presented in Ref. [98, 100]. Therefore, extra weight with the identity matrix (as WKGC<sup>2</sup> in Ref. [98]) is added to decrease the contribution of the SKGC formulation to eliminate the artifact, but it still shows considerable energy loss. Similarly, in the progressive wave propagation problem, compared to the result without correction [60], the low-resolution results obtained by the RKGC formulation already show great alleviation of the decay, and the high-resolution results don't show apparent decay, aligning with the theoretical amplitude envelope, indicating good energy conservation. But for the other attempts presented in Ref. [60], it can achieve similar aims; it requires great computational cost to obtain the inversion of the correction matrix for each particle pair at every time step. For the current method, with the adoption of dual-time stepping techniques [105], there is only an additional 10% and 15% computational cost for 2D and 3D, respectively, to calculate KGC matrices, indicating good efficiency.

Apart from the simulation, optimization using the SPH method also holds excellent advantages. For the PDE-constrained problems, specifically the thermal conductivity optimization considered in the current thesis, many simulated-based methods, either based on deterministic principles [70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80] or those stochastic approaches [83, 84, 85, 86], they often deal with a large number of variables in the spatial discretization of the domain, leading to reduced efficiency and suitability for large-scale and reliability-sensitive problems. However, in the SPH method, the particle only interacts with its neighbors, allowing it to divide the whole domain into manageable subdomain systems, potentially improving the optimization efficiency. In paper II, we introduce a target-driven all-at-once method for PDE-constrained optimization problems and derive a splitting SPH method for optimizing the thermal conductivity distribution to minimize the average temperature. The novelty of the method can be summarized in three aspects. Firstly, as the split steps are only weakly coupled with each other, compared to previous all-at-once approaches, the present updating of both state and design variables is much easier to handle. Secondly, by leveraging the splitting-operator SPH method, implicit updating is achieved without the inversion of large-size matrices. Thirdly, a general formulation of regularization has been proposed to achieve numerical stability when evolving the design variable. Typical examples of heat conduction optimization demonstrate that the current method yields optimal results comparable to previous methods and exhibits considerable computational efficiency. Moreover, the optimal results feature more moderate extreme values, which offers distinct advantages for the easier selection of appropriate material with high thermal conductivity. For example, regarding a thermal domain with a constant heat source and two heat sinks, the current method obtains the optimized temperature with the comparable reduction in temperature to the reference by AD [66], TGH [66], and AA [63] methods, but the present method is quite efficient, as it only takes a few times the computation cost of obtaining a steady-state solution to achieve the optimized result. Other examples all demonstrate a similar situation, indicating the greatly improved efficiency of optimization with the current method.

As the numerical algorithm for simulation and optimization based on the SPH is always developed in open-source libraries, modifications to the source code by

bug fixing or new feature development, should not compromise the accuracy and functionality already validated and verified. Regression test could ensure the output validity of scientific computing libraries under development, and it has been implemented in some mesh-based open-source libraries [131, 132, 133, 134, 31, 135], a testing environment for meshless libraries is notably absent, even as the popularity of meshless computing grows. Compared to the fixed topology of mesh-based methods, this gap can be attributed to the unique challenges posed by the Lagrangian property of the meshless methods, where dynamic and varying particle distributions in each computation make it challenging to establish standardized testing procedures. Notably, when updating the source code of SPHinXsys [30], an open-source multi-physics library developed by our group, we encountered instances where certain test cases passed the CTest (CMake Test) [136], yet unexpectedly resulted in simulation crashes without any error output. We had initially believed these test cases to be correct since they passed the testing, only to realize later that they were faulty. This issue proved troublesome and was the primary motivation for establishing an effective regression test environment. Therefore, in paper III, we introduce a methodological framework for constructing an automatic regression test environment in scientific computing libraries, and it has been successful in guaranteeing the consistent development of the library and improving the efficiency in simulation and optimization with reliable tools.

## 4.2 Outlooks

The present work can be further improved in several directions to further advance the consistency and efficiency of the SPH method, and some key points for further work are:

- While the RKGC formulation can be straightforwardly extended to even higher-order consistencies, challenges in achieving converged solutions for particle relaxation driven by the KGC matrix or correction functions are yet to be elaborated, especially for the three-dimensional complex geometries and the solution employing a  $h/\Delta x$  value smaller than 1.0. Therefore, breaking this limitation will not only further improve the numerical consistency of the SPH method but also enhance computational efficiency, as fewer neighbors in a reduced smoothing length can yield accurate results.
- In addition, the RKGC formulation has already been employed in the momentum conservation equation for the Lagrangian SPH method and both the mass and momentum conservation equations for the Eulerian SPH method and gained improved results. Its application for the Lagrangian continuity equation, as well as for the SPH solid dynamics still needs to be studied so that SPH can be developed into an efficient numerical method for general multi-physics simulations.
- The splitting SPH method is currently adopted to solve the thermal conductivity optimization problem with a target on the whole domain but minimizing hot spot temperature, i.e., the local target is also important in electronic cooling. Thus, the corresponding optimization algorithm based on the splitting SPH still needs further exploration. In addition, the continued distribution of thermal conductivity is challenging for manufacturing and engineering, so discrete distribution can also be studied to be practical friendly.

- The target-driven optimization concept used in the present thesis is not restricted to heat conduction optimization; it may be extended for topology optimization or fluid dynamics applications, such as drag reduction and lift maximum, to further improve the optimization efficiency in various applications.
- Finally, the current test in the SPHinXsys is limited. As the number of test cases grows due to the addition of new dynamics features, additional regression test techniques, such as the selection and reduction of test cases, must be conducted to make the test process more efficient.

# Bibliography

- [1] J. B. Drake, P. W. Jones, and G. R. Carr Jr. "Overview of the software design of the community climate system model". In: *The International Journal of High Performance Computing Applications* 19.3 (2005), pp. 177–186.
- [2] S. Ghosh and C. Misra. "Assessing hydrological impacts of climate change: modeling techniques and challenges". In: *The Open Hydrology Journal* 4.1 (2010).
- [3] D. E. Post and R. P. Kendall. "Software project management and quality engineering practices for complex, coupled multiphysics, massively parallel computational simulations: Lessons learned from ASCI". In: *The International Journal of High Performance Computing Applications* 18.4 (2004), pp. 399–416.
- [4] J. Smolka, O. Bíró, and A. J. Nowak. "Numerical simulation and experimental validation of coupled flow, heat transfer and electromagnetic problems in electrical transformers". In: *Archives of Computational Methods in Engineering* 16 (2009), pp. 319–355.
- [5] S. V. Paras and A. G. Kanaris. "Experimental and numerical studies in biomedical engineering". In: *Fluids* 4.2 (2019), p. 106.
- [6] L. K. Foong et al. "Numerical simulation of blood flow inside an artery under applying constant heat flux using Newtonian and non-Newtonian approaches for biomedical engineering". In: *Computer methods and programs in biomedicine* 190 (2020), p. 105375.
- [7] B. Mi and H. Zhan. "Numerical simulation of the static and dynamic aerodynamics of a UAV under wake flows". In: *Journal of Advanced Transportation* 2019.1 (2019), p. 6326794.
- [8] M. Bras, S. Warwick, and A. Suleman. "Aeroelastic evaluation of a flexible high aspect ratio wing UAV: Numerical simulation and experimental flight validation". In: *Aerospace Science and Technology* 122 (2022), p. 107400.
- [9] M. A. Muflikhun and T. Yokozeki. "Experimental and numerical analysis of CFRP-SPCC hybrid laminates for automotive and structural applications with cost analysis assessment". In: *Composite Structures* 263 (2021), p. 113707.
- [10] S. A. Prabhakaran, G. Balaji, and K. Annamalai. "Numerical simulation of crashworthiness parameters for design optimization of an automotive crash-box". In: *International Journal for Simulation and Multidisciplinary Design Optimization* 13 (2022), p. 3.
- [11] J. C. De los Reyes. "Numerical PDE-constrained optimization. SpringerBriefs in Optimization". In: *Springer, Cham. doi 10* (2015), pp. 978–3.
- [12] A. L. Moore and L. Shi. "Emerging challenges and materials for thermal management of electronics". In: *Materials today* 17.4 (2014), pp. 163–174.
- [13] M. Almgöbel and A. Bejan. "Conduction trees with spacings at the tips". In: *International Journal of Heat and Mass Transfer* 42.20 (1999), pp. 3739–3756.

- [14] Z. Y. Guo, X. G. Cheng, and Z. Z. Xia. "Least dissipation principle of heat transport potential capacity and its application in heat conduction optimization". In: *Chinese science bulletin* 48 (2003), pp. 406–410.
- [15] J. Anderson. *EBOOK: Fundamentals of Aerodynamics (SI units)*. McGraw hill, 2011.
- [16] D. I. Papadimitriou and C. Papadimitriou. "Aerodynamic shape optimization for minimum robust drag and lift reliability constraint". In: *Aerospace Science and Technology* 55 (2016), pp. 24–33.
- [17] L. Leifsson and S. Koziel. "Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models: a review of recent progress". In: *Journal of Computational Science* 10 (2015), pp. 45–54.
- [18] J. D. Anderson and J. Wendt. *Computational fluid dynamics*. Vol. 206. Springer, 1995.
- [19] O. C. Zienkiewicz and R. L. Taylor. *The finite element method: solid mechanics*. Vol. 2. Butterworth-heinemann, 2000.
- [20] R. Eymard, T. Gallouët, and R. Herbin. "Finite volume methods". In: *Handbook of numerical analysis* 7 (2000), pp. 713–1018.
- [21] L. 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.
- [22] R. A. Gingold and J. J. Monaghan. "Smoothed particle hydrodynamics: theory and application to non-spherical stars". In: *Monthly notices of the royal astronomical society* 181.3 (1977), pp. 375–389.
- [23] B. K. Mishra and R. K. Rajamani. "The discrete element method for the simulation of ball mills". In: *Applied Mathematical Modelling* 16.11 (1992), pp. 598–604.
- [24] S. Koshizuka and Y. Oka. "Moving-particle semi-implicit method for fragmentation of incompressible fluid". In: *Nuclear science and engineering* 123.3 (1996), pp. 421–434.
- [25] R. Herzog. "Lectures notes algorithms and preconditioning in pde-constrained optimization". In: *Technical Report* (2010).
- [26] R. Herzog and K. Kunisch. "Algorithms for PDE-constrained optimization". In: *GAMM-Mitteilungen* 33.2 (2010), pp. 163–176.
- [27] E. Madenci and I. Guven. *The finite element method and applications in engineering using ANSYS®*. Springer, 2015.
- [28] R. W. Pryor. *Multiphysics modeling using COMSOL®: a first principles approach*. Jones & Bartlett Publishers, 2009.
- [29] H. Jasak, A. Jemcov, and Z. Tukovic. "OpenFOAM: A C++ library for complex physics simulations". In: *International workshop on coupled methods in numerical dynamics*. Vol. 1000. Dubrovnik, Croatia). 2007, pp. 1–20.
- [30] C. 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.
- [31] P. E. Farrell et al. "Automated continuous verification for numerical simulation". In: *Geoscientific Model Development* 4.2 (2011), pp. 435–449.

- [32] M. B. Liu and G. R. Liu. "Smoothed particle hydrodynamics (SPH): an overview and recent developments". In: *Archives of computational methods in engineering* 17 (2010), pp. 25–76.
- [33] T. Ye et al. "Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications". In: *Physics of Fluids* 31.1 (2019).
- [34] C. Zhang et al. "Review on smoothed particle hydrodynamics: Methodology development and recent achievement". In: *arXiv preprint arXiv:2205.03074* (2022).
- [35] R. A. Gingold and J. J. Monaghan. "Smoothed particle hydrodynamics: theory and application to non-spherical stars". In: *Monthly notices of the royal astronomical society* 181.3 (1977), pp. 375–389.
- [36] J. J. Monaghan. "Simulating free surface flows with SPH". In: *Journal of computational physics* 110.2 (1994), pp. 399–406.
- [37] X. Y. Hu and N. A. Adams. "A multi-phase SPH method for macroscopic and mesoscopic flows". In: *Journal of Computational Physics* 213.2 (2006), pp. 844–861.
- [38] C. Zhang, X. Y. Hu, and N. A. Adams. "A weakly compressible SPH method based on a low-dissipation Riemann solver". In: *Journal of Computational Physics* 335 (2017), pp. 605–620.
- [39] W. Benz and E. Asphaug. "Simulations of brittle solids using smooth particle hydrodynamics". In: *Computer physics communications* 87.1-2 (1995), pp. 253–265.
- [40] T. Rabczuk and J. Eibl. "Simulation of high velocity concrete fragmentation using SPH/MLSPH". In: *International Journal for Numerical Methods in Engineering* 56.10 (2003), pp. 1421–1444.
- [41] Y. X. Peng et al. "A meshfree framework for the numerical simulation of elasto-plasticity deformation of ship structure". In: *Ocean Engineering* 192 (2019), p. 106507.
- [42] H. Gotoh and A. Khayyer. "On the state-of-the-art of particle methods for coastal and ocean engineering". In: *Coastal Engineering Journal* 60.1 (2018), pp. 79–103.
- [43] C. Zhang, M. Rezavand, and X. Y. Hu. "A multi-resolution SPH method for fluid-structure interactions". In: *Journal of Computational Physics* 429 (2021), p. 110028.
- [44] D. A. Fulk. *A numerical analysis of smoothed particle hydrodynamics*. Air Force Institute of Technology, 1994.
- [45] D. A. Fulk and D. W. Quinn. "An analysis of 1-D smoothed particle hydrodynamics kernels". In: *Journal of Computational Physics* 126.1 (1996), pp. 165–180.
- [46] N. J. Quinlan, M. Basa, and M. Lastiwka. "Truncation error in mesh-free particle methods". In: *International Journal for Numerical Methods in Engineering* 66.13 (2006), pp. 2064–2085.
- [47] S. Litvinov, X. Y. Hu, and N. A. Adams. "Towards consistence and convergence of conservative SPH approximations". In: *Journal of Computational Physics* 301 (2015), pp. 394–401.

- [48] J. J. Monaghan. "Smoothed particle hydrodynamics". In: *Annual review of astronomy and astrophysics* 30.1 (1992), pp. 543–574.
- [49] V. G. Maz'ia and G. Schmidt. *Approximate approximations*. 141. American Mathematical Soc., 2007.
- [50] G. Oger et al. "An improved SPH method: Towards higher order convergence". In: *Journal of Computational Physics* 225.2 (2007), pp. 1472–1492.
- [51] P. W. Randles and L. D. Libersky. "Smoothed particle hydrodynamics: some recent improvements and applications". In: *Computer methods in applied mechanics and engineering* 139.1-4 (1996), pp. 375–408.
- [52] J. K. Chen, J. E. Beraun, and T. C. Carney. "A corrective smoothed particle method for boundary value problems in heat conduction". In: *International Journal for Numerical Methods in Engineering* 46.2 (1999), pp. 231–252.
- [53] W. K. Liu, S. Jun, and Y. F. Zhang. "Reproducing kernel particle methods". In: *International journal for numerical methods in fluids* 20.8-9 (1995), pp. 1081–1106.
- [54] W. K. Liu et al. "Reproducing kernel particle methods for structural dynamics". In: *International Journal for Numerical Methods in Engineering* 38.10 (1995), pp. 1655–1679.
- [55] S. N. Atluri, J. Y. Cho, and H. G. Kim. "Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations". In: *Computational mechanics* 24.5 (1999), pp. 334–347.
- [56] A. M. A. 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.
- [57] J. Bonet and T. S. L. Lok. "Variational and momentum preservation aspects of smooth particle hydrodynamic formulations". In: *Computer Methods in applied mechanics and engineering* 180.1-2 (1999), pp. 97–115.
- [58] P. R. R. de Campos et al. "A New Updated Reference Lagrangian Smooth Particle Hydrodynamics algorithm for isothermal elasticity and elasto-plasticity". In: *Computer Methods in Applied Mechanics and Engineering* 392 (2022), p. 114680.
- [59] J. P. Vila. "SPH renormalized hybrid methods for conservation laws: applications to free surface flows". In: *Meshfree methods for partial differential equations II*. Springer, 2005, pp. 207–229.
- [60] V. Zago et al. "Overcoming excessive numerical dissipation in SPH modeling of water waves". In: *Coastal Engineering* 170 (2021), p. 104018.
- [61] G. Q. Liang et al. "Study on the propagation of regular water waves in a numerical wave flume with the  $\delta$ -SPHC model". In: *Applied Ocean Research* 135 (2023), p. 103559.
- [62] K. Ito and K. Kunisch. *Lagrange multiplier approach to variational problems and applications*. SIAM, 2008.
- [63] T. Zhao, X. Wu, and Z. Y. Guo. "Optimal thermal conductivity design for the volume-to-point heat conduction problem based on adjoint analysis". In: *Case Studies in Thermal Engineering* 40 (2022), p. 102471.
- [64] L. B. Rall and G. F. Corliss. "An introduction to automatic differentiation". In: *Computational Differentiation: Techniques, Applications, and Tools* 89 (1996), pp. 1–18.

- [65] O. I. Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4.11 (2018).
- [66] M. X. Song, K. Chen, and X. Zhang. "Optimization of the volume-to-point heat conduction problem with automatic differentiation based approach". In: *International Journal of Heat and Mass Transfer* 177 (2021), p. 121552.
- [67] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- [68] W. Alt and K. Malanowski. "The Lagrange-Newton method for nonlinear optimal control problems". In: *Computational Optimization and Applications* 2 (1993), pp. 77–100.
- [69] A. Bejan. "Constructal-theory network of conducting paths for cooling a heat generating volume". In: *International Journal of Heat and Mass Transfer* 40.4 (1997), pp. 799–816.
- [70] X. G. Cheng, Z. X. Li, and Z. Y. Guo. "Constructs of highly effective heat transport paths by bionic optimization". In: *Science in China Series E: Technological Sciences* 46 (2003), pp. 296–302.
- [71] X. T. Cheng, X. H. Xu, and X. G. Liang. "Homogenization of temperature field and temperature gradient field". In: *Science in China Series E: Technological Sciences* 52 (2009), pp. 2937–2942.
- [72] Z. Y. Guo, H. Y. Zhu, and X. G. Liang. "Entransy—A physical quantity describing heat transfer ability". In: *International Journal of Heat and Mass Transfer* 50.13-14 (2007), pp. 2545–2556.
- [73] A. Bejan and J. Kestin. "Entropy generation through heat and fluid flow". In: (1983).
- [74] A. Bejan. "Entropy generation minimization: The new thermodynamics of finite-size devices and finite-time processes". In: *Journal of Applied Physics* 79.3 (1996), pp. 1191–1218.
- [75] W. J. Du et al. "Optimization of volume to point conduction problem based on a novel thermal conductivity discretization algorithm". In: *Chinese Journal of Chemical Engineering* 23.7 (2015), pp. 1161–1168.
- [76] R. Wang et al. "Constructal design of elliptical cylinders with heat generating for entropy generation minimization". In: *Entropy* 22.6 (2020), p. 651.
- [77] Q. Chen, X. G. Liang, and Z. Y. Guo. "Entransy theory for the optimization of heat transfer—a review and update". In: *International Journal of Heat and Mass Transfer* 63 (2013), pp. 65–81.
- [78] T. Zhao, Y. C. Hua, and Z. Y. Guo. "Irreversibility evaluation for transport processes revisited". In: *International Journal of Heat and Mass Transfer* 189 (2022), p. 122699.
- [79] T. Zhao, D. Liu, and Q. Chen. "A collaborative optimization method for heat transfer systems based on the heat current method and entransy dissipation extremum principle". In: *Applied Thermal Engineering* 146 (2019), pp. 635–647.
- [80] S. J. Wu et al. "Study on the performance of a miniscale channel heat sink with Y-shaped unit channels based on entransy analysis". In: *Applied Thermal Engineering* 209 (2022), p. 118295.
- [81] Z. X. Tong et al. "Optimizing thermal conductivity distribution for heat conduction problems with different optimization objectives". In: *International Journal of Heat and Mass Transfer* 119 (2018), pp. 343–354.

- [82] J. J. Zhang et al. "An effective method for hot spot temperature optimization in heat conduction problem". In: *Applied Thermal Engineering* (2023), p. 120325.
- [83] Z. Z. Xia, Z. X. Li, and Z. Guo. "Heat conduction optimization: high conductivity constructs based on the principle of biological evolution". In: *International Heat Transfer Conference Digital Library*. Begel House Inc. 2002.
- [84] Z. Z. Xia et al. "Bionic optimization of heat transport paths for heat conduction problems". In: *Journal of Enhanced Heat Transfer* 11.2 (2004).
- [85] R. Boichot, L. G. Luo, and Y. L. Fan. "Tree-network structure generation for heat conduction by cellular automaton". In: *Energy Conversion and Management* 50.2 (2009), pp. 376–386.
- [86] R. Boichot and L. G. Luo. "A simple cellular automaton algorithm to optimise heat transfer in complex configurations". In: *International Journal of Exergy* 7.1 (2010), pp. 51–64.
- [87] X. H. Xu, X. G. Liang, and J. X. Ren. "Optimization of heat conduction using combinatorial optimization algorithms". In: *International journal of heat and mass transfer* 50.9-10 (2007), pp. 1675–1682.
- [88] G. Bilotta et al. "GPUSPH: a Smoothed Particle Hydrodynamics model for the thermal and rheological evolution of lava flows". In: *Geological Society, London, Special Publications* 426.1 (2016), pp. 387–408.
- [89] M. Gomez-Gesteira et al. "SPHysics – development of a free-surface fluid solver – Part 1: Theory and formulations". In: *Computers and Geosciences* 48 (2012), pp. 289–299. ISSN: 0098-3004.
- [90] A. J. C. Crespo et al. "DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)". In: *Computer Physics Communications* 187 (2015), pp. 204–216.
- [91] J. L. Cercos-Pita. "AQUA<sub>gpusph</sub>, a new free 3D SPH solver accelerated with OpenCL". In: *Computer Physics Communications* 192 (2015), pp. 295–312.
- [92] V. Springel. "The cosmological simulation code GADGET-2". In: *Monthly notices of the royal astronomical society* 364.4 (2005), pp. 1105–1134.
- [93] P. F. Hopkins. "A new public release of the GIZMO code". In: *arXiv preprint arXiv:1712.01294* (2017).
- [94] B. Zhang, C. Zhang, and X. Y. Hu. "Target-driven splitting SPH optimization of thermal conductivity distribution". In: *International Journal of Heat and Mass Transfer* 227 (2024), p. 125476.
- [95] C. Zhang et al. "A weakly compressible SPH method with WENO reconstruction". In: *Journal of Computational Physics* 392 (2019), pp. 1–18.
- [96] C. Zhang, X. Y. Hu, and N. A. Adams. "A generalized transport-velocity formulation for smoothed particle hydrodynamics". In: *Journal of Computational Physics* 337 (2017), pp. 216–232.
- [97] J. J. Monaghan. "SPH and Riemann solvers". In: *Journal of Computational Physics* 136.2 (1997), pp. 298–307.
- [98] Y. R. Ren et al. "An efficient correction method in Riemann SPH for the simulation of general free surface flows". In: *Computer Methods in Applied Mechanics and Engineering* 417 (2023), p. 116460.

- [99] X. T. Huang et al. "Development of a numerical wave tank with a corrected smoothed particle hydrodynamics scheme to reduce nonphysical energy dissipation". In: *Chinese Journal of Theoretical and Applied Mechanics* 54.6 (2022), pp. 1502–1515.
- [100] H. G. Lyu et al. "Derivation of an improved smoothed particle hydrodynamics model for establishing a three-dimensional numerical wave tank overcoming excessive numerical dissipation". In: *Physics of Fluids* 35.6 (2023).
- [101] J. J. Monaghan. "Smoothed particle hydrodynamics". In: *Reports on progress in physics* 68.8 (2005), p. 1703.
- [102] D. J. Price. "Modelling discontinuities and Kelvin–Helmholtz instabilities in SPH". In: *Journal of Computational Physics* 227.24 (2008), pp. 10040–10057.
- [103] A. Colagrossi, M. Antuono, and D. Le Touzé. "Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model". In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 79.5 (2009), p. 056701.
- [104] A. Colagrossi and M. Landrini. "Numerical simulation of interfacial flows by smoothed particle hydrodynamics". In: *Journal of computational physics* 191.2 (2003), pp. 448–475.
- [105] C. Zhang, M. Rezavand, and X.Y. Hu. "Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics". In: *Journal of Computational Physics* 404 (2020), p. 109135.
- [106] J. P. Vila. "On particle weighted methods and smooth particle hydrodynamics". In: *Mathematical models and methods in applied sciences* 9.02 (1999), pp. 161–209.
- [107] Z. T. Wang et al. "Extended Eulerian SPH and its realization of FVM". In: *arXiv preprint arXiv:2309.01596* (2023).
- [108] A. Harten, P. D. Lax, and B. Leer. "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws". In: *SIAM review* 25.1 (1983), pp. 35–61.
- [109] F. Rieber. "On the dissipation mechanism of upwind-schemes in the low Mach number regime: A comparison between Roe and HLL". In: *Journal of Computational Physics* 229.2 (2010), pp. 221–232.
- [110] S. Adami, X. Y. Hu, and N. A. Adams. "A generalized wall boundary condition for smoothed particle hydrodynamics". In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075.
- [111] E. M. Ryan, A. M. Tartakovsky, and C. Amon. "A novel method for modeling Neumann and Robin boundary conditions in smoothed particle hydrodynamics". In: *Computer Physics Communications* 181.12 (2010), pp. 2008–2023.
- [112] Y. J. 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.
- [113] S. Adami, X. Y. Hu, and N. A. Adams. "A transport-velocity formulation for smoothed particle hydrodynamics". In: *Journal of Computational Physics* 241 (2013), pp. 292–307.
- [114] G. Strang. "On the construction and comparison of difference schemes". In: *SIAM journal on numerical analysis* 5.3 (1968), pp. 506–517.

- [115] K. Nguyen, A. Caboussat, and D. Dabdub. "Mass conservative, positive definite integrator for atmospheric chemical dynamics". In: *Atmospheric Environment* 43.40 (2009), pp. 6287–6295.
- [116] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.
- [117] Y. J. Zhu, C. Zhang, and X. Hu. "A dynamic relaxation method with operator splitting and random-choice strategy for SPH". In: *Journal of Computational Physics* 458 (2022), p. 111105.
- [118] R. Bagnara. "A unified proof for the convergence of Jacobi and Gauss–Seidel methods". In: *SIAM review* 37.1 (1995), pp. 93–97.
- [119] R. C. Batra and G. M. Zhang. "Analysis of adiabatic shear bands in elasto-thermo-viscoplastic materials by modified smoothed-particle hydrodynamics (MSPH) method". In: *Journal of computational physics* 201.1 (2004), pp. 172–190.
- [120] G. M. Zhang and R. C. Batra. "Modified smoothed particle hydrodynamics method and its application to transient problems". In: *Computational mechanics* 34.2 (2004), pp. 137–146.
- [121] S. Sibilla. "An algorithm to improve consistency in smoothed particle hydrodynamics". In: *Computers & Fluids* 118 (2015), pp. 148–158.
- [122] A. M. A. Nasar et al. "High-order consistent SPH with the pressure projection method in 2-D and 3-D". In: *Journal of Computational Physics* 444 (2021), p. 110563.
- [123] B. Zhang, N. A. Adams, and X. Y. Hu. "Towards high-order consistency and convergence of conservative SPH approximations". In: *Computer Methods in Applied Mechanics and Engineering* 433 (2025), p. 117484.
- [124] H. Remmel et al. "System testing a scientific framework using a regression-test environment". In: *Computing in Science & Engineering* 14.2 (2011), pp. 38–45.
- [125] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49.
- [126] C. Myers, L. Rabiner, and A. Rosenberg. "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.6 (1980), pp. 623–635.
- [127] C. Zhang et al. "A simple artificial damping method for total Lagrangian smoothed particle hydrodynamics". In: *arXiv preprint arXiv:2102.04898* (2021).
- [128] C. Zhang et al. "An integrative smoothed particle hydrodynamics method for modeling cardiac function". In: *Computer Methods in Applied Mechanics and Engineering* 381 (2021), p. 113847.
- [129] B. Zhang, C. Zhang, and X. Y. Hu. "Automated regression test method for scientific computing libraries: Illustration with SPHinXsys". In: *Journal of Hydrodynamics* (2024), pp. 1–13.
- [130] A. Khayyer et al. "Enhanced resolution of the continuity equation in explicit weakly compressible SPH simulations of incompressible free-surface fluid flows". In: *Applied Mathematical Modelling* 116 (2023), pp. 84–121.

- 
- [131] X. Y. Lin, M. Simon, and N. Niu. “Exploratory metamorphic testing for scientific software”. In: *Computing in science & engineering* 22.2 (2018), pp. 78–87.
  - [132] X. Y. Lin, M. Simon, and N. Niu. “Hierarchical metamorphic relations for testing scientific software”. In: *2018 IEEE/ACM 13th International Workshop on Software Engineering for Science (SE4Science)*. IEEE. 2018, pp. 1–8.
  - [133] Z. D. Peng, X. Lin, and N. Niu. “Unit tests of scientific software: A study on SWMM”. In: *International Conference on Computational Science*. Springer. 2020, pp. 413–427.
  - [134] Z. D. Peng et al. “Unit and regression tests of scientific software: A study on SWMM”. In: *Journal of Computational Science* 53 (2021), p. 101347.
  - [135] H. J. Happ. “Linux C-Shell Regression Testing for the SHAMRC CFD Code”. In: *Users’ Group Conference*. 2011, p. 41.
  - [136] *Kitware Inc. et al., CMake*. <https://cmake.org/documentation/>, accessed on 7th July 2021.



## Appendix A

# Original journal papers

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



## A.1 Paper I

Bo Zhang, Nikolaus Adams, Xiangyu Hu

### **Towards high-order consistency and convergence of conservative SPH approximations**

In *Computer Methods in Applied Mechanics and Engineering*, Volume 433: 117484, 2025,  
DOI: <https://doi.org/10.1016/j.cma.2024.117484>.

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.





# Towards high-order consistency and convergence of conservative SPH approximations

Bo Zhang, Nikolaus Adams, Xiangyu Hu\*

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

## ARTICLE INFO

### Keywords:

Smoothed particle hydrodynamics (SPH)  
Reverse kernel gradient correction (RKGC)  
Conservative approximation  
First-order consistency  
Particle relaxation  
Transport-velocity formulation

## ABSTRACT

Smoothed particle hydrodynamics (SPH) offers distinct advantages for modeling many engineering problems, yet achieving high-order consistency in its conservative formulation remains to be addressed. While zero- and higher-order consistencies can be obtained using particle-pair differences and kernel gradient correction (KGC) approaches, respectively, for SPH gradient approximations, their applicability for discretizing conservation laws in practical simulations is limited due to their lack of discrete conservation. Although the standard anti-symmetric SPH approximation is able to achieve conservation and zero-order consistency through particle relaxation, its straightforward extensions with the KGC fail to satisfy zero- or higher-order consistency. In this paper, we propose the reverse KGC (RKGC) formulation, which is conservative and able to satisfy up to first-order consistency when particles are relaxed based on the KGC matrix. Extensive numerical tests show that the new formulation considerably improves the accuracy of the Lagrangian SPH method. In particular, it is able to resolve the long-standing high-dissipation issue for simulating free-surface flows. Furthermore, with fully relaxed particles, it enhances the accuracy of the Eulerian SPH method even when the ratio between the smoothing length and the particle spacing is considerably reduced. The reverse KGC formulation holds the potential for extension to even higher-order consistencies with a pending challenge in addressing the corresponding particle relaxation problem.

## 1. Introduction

As a mesh-free method, smoothed particle hydrodynamics (SPH), initially proposed by Lucy [1] and Gingold & Monaghan [2] for astrophysical applications, has demonstrated significant success across a wide range of scientific problems. These include fluid dynamics [3–5], solid dynamics [6–8], and fluid–structure interaction [9–11], among others. The SPH approximation operates on the principle of reconstructing the continuous field and its spatial derivatives from a set of discrete particles, each possessing individual properties, through a Gaussian-like smoothing kernel function with compact support [12,13]. It generally encounters two different types of errors that amalgamate to the overall truncation error [14,15]. The first is the smoothing error determined by the kernel function where the leading moments vanish. This error arises due to the discrepancy between the smoothing approximations and the exact values. The second is the integration error, characterized by the non-vanishing of leading moments due to the particle approximation. For typical SPH kernel functions, such as the cubic B-spline [16] and the Wendland kernel [17] that are second-order accuracy, given that only the first moment vanishes, corresponding to first-order consistency. For a given kernel function, Quinlan et al. [14] observed that the overall truncation error generally decreases or exhibits consistency with increased resolution only when  $h/\Delta x$ , i.e., the ratio between the smoothing length  $h$  and the particle spacing  $\Delta x$ , is large, suggesting sufficient small integration

\* Corresponding author.

E-mail addresses: [bo.zhang.aer@tum.de](mailto:bo.zhang.aer@tum.de) (B. Zhang), [nikolaus.adams@tum.de](mailto:nikolaus.adams@tum.de) (N. Adams), [xiangyu.hu@tum.de](mailto:xiangyu.hu@tum.de) (X. Hu).

<https://doi.org/10.1016/j.cma.2024.117484>

Received 24 June 2024; Received in revised form 25 September 2024; Accepted 17 October 2024

0045-7825/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

**Acronyms**

<i>BR</i>	Correction-matrix based particle relaxation
<i>BT</i>	Correction-matrix based transport velocity formulation
<i>ESPH</i>	Eulerian SPH
<i>KGC</i>	Kernel gradient correction in non-conservative form
<i>NKGC</i>	Without KGC in conservative form
<i>PR</i>	Pressure based particle relaxation
<i>PT</i>	Pressure based transport velocity formulation
<i>RKGC</i>	Reverse KGC in conservative form
<i>SKGC</i>	Straightforward KGC in conservative form
<i>WCSPH</i>	Weakly compressible SPH
<i>WKGC</i> <sup>1</sup>	Weighted KGC in conservative form (only for free-surface)
<i>WKGC</i> <sup>2</sup>	Weighted KGC in conservative form (for the whole domain)

error. However, achieving this condition leads to an excessive number of particles within the kernel compact support and results in an extremely high computational cost.

Therefore, different approaches have been proposed to minimize integration error or improve consistency with a computationally acceptable value of  $h/\Delta x$ , typically less than 1.5 [18]. Besides that zero-order consistency can be easily achieved by using particle-pair differences for gradient approximations, high-order consistencies can be attained through kernel gradient correction (KGC) [19] and various similar approaches, such as corrective smoothed particle method (CSPM) [20], reproducing kernel particle method (RKPM) [21,22], moving least squared (MLS) [23], finite particle method (FPM) [24–29], modified SPH method (MSPH) [30–33], kernel gradient free (KGF) [34,35], and many others [36–38]. While these approaches are able to achieve consistencies for the SPH approximation of the gradient and/or Laplacian operators and improve accuracy, their application to the discretization of physical conservation laws still faces the significant challenge of non-conservation. Specifically, conservative discretization necessitates an anti-symmetric form between particle pairs, a condition that these methods cannot appropriately satisfy.

The first approach to achieve both zero-order consistency and conservation involves implementing particle relaxation based on a constant background pressure [15,39] before applying anti-symmetric SPH approximations. Since particle relaxation is computationally expensive for the Lagrangian SPH method, an alternative one-step consistency correction, such as the transport-velocity formulation [40,41], based on the same principle, has also been proposed to enhance the consistency. Although conservation properties always hold a high priority [18], there is still a need for high-order consistency in SPH to ensure truncation-errors decay with increase particle resolution. To address this issue, approaches have been developed to enhance numerical accuracy and consistency by integrating KGC with anti-symmetric formulations [42–44] and/or particle relaxation [45]. These methods include utilizing the average correction matrix [46–48] and implementing separate corrections for each particle pair [18,49], etc. For some problems, while these conservative KGC formulations have demonstrated improved results compared to standard SPH methods without KGC, as will be shown later, they not only fail to achieve high-order consistency but also lose zero-order consistency, even with particle relaxation.

In this study, we introduce the reverse KGC (RKGC) formulation designed for conservative SPH approximations. In contrast to prior conservative KGC formulations, RKGC is able to precisely fulfill both zero- and first-order consistencies of the gradient operator. The formulation incorporates a particle relaxation driven by the KGC matrix and is split into two parts: the first part addresses zero-order consistency and vanishes during particle relaxation, while the second part ensures first-order consistency and accurately reproduces linear gradients. The formulation notably improves numerical accuracy in Lagrangian SPH simulations. In particular, it exhibits very good energy conservation properties and resolves the long-standing high-dissipation issue for the SPH simulation of free-surface flows. Moreover, since these consistencies can be strictly imposed in the Eulerian SPH method, even when employing a reduced smoothing length, the formulation still has the potential to yield results with improved accuracy.

In the following, Section 2 presents the approximation of gradients in the SPH method and its application in discretizing governing equations. Section 3 introduces the RKGC formulation, detailing the KGC-based particle relaxation and the transport-velocity formulation, and then the variational consistency analysis. Section 4 conducts corresponding error and convergence analyses. Section 5 presents extensive numerical examples that highlight the benefits gained from the proposed method. In Section 6, we extend the RKGC formulation to second-order consistency, with a yet-to-be-addressed condition on particle relaxation. Finally, Section 7 summarizes the key findings and outlines of future research.

## 2. Preliminary

### 2.1. Gradient approximation

In SPH discretization of the partial differential equation of fluid dynamics, the kernel approximation for the gradient operator acting on a smooth field  $\psi(\mathbf{r})$  can be expressed through a two-stage approach

$$\nabla\psi(\mathbf{r}) \approx \int_{\Omega} \nabla\psi(\mathbf{r}^*) W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = - \int_{\Omega} \psi(\mathbf{r}^*) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*, \quad (1)$$

where  $W(\mathbf{r}, h)$  is the kernel function scaled by the smoothing length  $h$ . While the first stage introduces smoothing errors by the kernel function, the second stage entails integration by parts, assuming the kernel function vanishes at the boundary of a compact support. Through Taylor expansion, for Eq. (1) one can find that the zero-order consistency condition is

$$\int_{\Omega} \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = 0, \quad (2)$$

and the first-order consistency condition is

$$-\int_{\Omega} (\mathbf{r}^* - \mathbf{r}) \otimes \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* = \mathbf{I}, \quad (3)$$

where  $\mathbf{I}$  represents the identity matrix. Zero-order consistency condition allows rewriting the kernel approximation in two equivalent forms:

$$\begin{aligned} \nabla \psi(\mathbf{r}) &= \int_{\Omega} (\psi(\mathbf{r}) - \psi(\mathbf{r}^*)) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^* \\ &\equiv -\int_{\Omega} (\psi(\mathbf{r}) + \psi(\mathbf{r}^*)) \nabla W(\mathbf{r} - \mathbf{r}^*, h) d\mathbf{r}^*. \end{aligned} \quad (4)$$

By introducing particle summation, the first approximation in Eq. (4) can be further approximated for an SPH particle  $i$  as

$$\nabla \psi_i = \sum_j \psi_{ij} \nabla W_{ij} V_j, \quad (5)$$

where  $V_j$  is the volume of the neighbor particles within the support, and the particle-pair difference is  $\psi_{ij} = \psi_i - \psi_j$ . This form is often referred to as a symmetric or non-conservative form. Similarly, the second approximation in Eq. (4) can be also approximated as

$$\nabla \psi_i = -\sum_j (\psi_i + \psi_j) \nabla W_{ij} V_j, \quad (6)$$

where the particle-pair sum is employed. This form, known as the anti-symmetric or conservative form, ensures discrete conservation and is commonly chosen in classical SPH methods for discretizing physical conservation laws.

For the non-conservative form of Eq. (5), zero-order consistency is automatically satisfied as the particle-pair difference is used. To achieve first-order consistency, one requires that the approximation of Eq. (3) satisfies

$$-\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j = \mathbf{I}. \quad (7)$$

To precisely fulfill the above condition, the KGC approach [19], introducing a correction matrix  $\mathbf{B}_i$  to adjust the gradient of the kernel function, can be employed, so that one has

$$-\sum_j \mathbf{r}_{ij} \otimes \mathbf{B}_i \nabla W_{ij} V_j = \mathbf{I}, \quad \mathbf{B}_i = \left( -\sum_j \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j \right)^{-1}. \quad (8)$$

With the KGC, Eq. (5) is modified as

$$\nabla \psi_i = \sum_j \psi_{ij} \mathbf{B}_i \nabla W_{ij} V_j. \quad (9)$$

Note that, introducing  $\mathbf{B}_i$  does not affect the zero-order consistency of Eq. (9). Also note that, although the non-conservation form is not desirable for the discretization of physical conservation laws, Eq. (9) is often used when the conservation is not a primary concern because it can reproduce the linear gradient and achieve second-order accuracy.

In the conservative form of Eq. (6), where the particle-pair sum other than the difference is used, the zero-order consistency condition becomes nontrivial as

$$\sum_j \nabla W_{ij} V_j = 0. \quad (10)$$

Litvinov et al. [15] proposed a particle relaxation process driven by a constant background pressure assuming invariant particle volume. After the particles are settled down or fully relaxed, Eq. (10) is satisfied for the zero-order consistency. To achieve first-order consistency, as a straightforward extension for Eq. (6), one may suggest

$$\nabla \psi_i = -\sum_j (\psi_i \mathbf{B}'_i + \psi_j \mathbf{B}'_j) \nabla W_{ij} V_j, \quad (11)$$

where  $\mathbf{B}'_i$  and  $\mathbf{B}'_j$  are some correction matrices for particles  $i$  and  $j$ , is able to reproduce a linear gradient similar to Eq. (9). How to obtain these correction matrices is not straightforward, and various attempts based on the original KGC matrix for the non-conservative form have been carried out. One widely used formulation, introduced by Oger et al. [18], is expressed as,

$$\nabla \psi_i = -\sum_j (\psi_i \mathbf{B}_i + \psi_j \mathbf{B}_j) \nabla W_{ij} V_j, \quad (12)$$

where the KGC matrix is applied for each particle pair separately.

## 2.2. Weakly compressible SPH (WCSPH)

The governing equations in the Lagrangian framework for viscous flows consist of the mass and momentum conservation equations, written as

$$\begin{cases} \frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \\ \frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g}, \end{cases} \quad (13)$$

where  $\rho$  represents density,  $\mathbf{v}$  velocity,  $p$  pressure,  $\nu$  kinematic viscosity,  $\mathbf{g}$  gravity and  $d(\bullet)/dt = \partial(\bullet)/\partial t + \mathbf{v} \cdot \nabla(\bullet)$  refers to the material derivative. An artificial equation of state (EOS) for weakly compressible flows is used to close Eq. (13) as

$$p = c_0^2 (\rho - \rho_0). \quad (14)$$

Here,  $\rho_0$  is the initial density, and  $c_0$  denotes the artificial sound speed. Setting  $c_0 = 10U_{max}$ , where  $U_{max}$  represents the anticipated maximum fluid speed, fulfills the weakly compressible assumption where the density variation remains around 1%.

The Riemann-SPH method [5] is employed here to discretize Eq. (13). Subsequently, the continuity and momentum equations are approximated as

$$\begin{cases} \frac{d\rho_i}{dt} = 2\rho_i \sum_j (\mathbf{v}_i - \mathbf{v}^*) \cdot \nabla W_{ij} V_j \\ \frac{d\mathbf{v}_i}{dt} = -\frac{2}{m_i} \sum_j P^* \nabla W_{ij} V_i V_j + 2 \sum_j \frac{\nu}{\rho_i} \frac{\mathbf{v}_{ij}}{r_{ij}} \frac{\partial W}{\partial r_{ij}} V_j + \mathbf{g}_i, \end{cases} \quad (15)$$

where  $\nabla W_{ij} = \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}$ , and  $\mathbf{e}_{ij} = \frac{\mathbf{r}_{ij}}{r_{ij}}$  is the unit vector. The particle-pair velocity  $\mathbf{v}^*$  and pressure  $P^*$ , respectively, are solutions obtained from the Riemann problem constructed along the interacting line of each pair of particles. Note that the particle-pair pressure  $P^*$  leads to an anti-symmetric form and hence momentum conservation. With a linearized Riemann solver, the solutions can be computed as

$$\begin{cases} \mathbf{v}^* = \bar{\mathbf{v}}_{ij} + (U^* - \bar{U}_{ij}) \mathbf{e}_{ij}, & U^* = \bar{U}_{ij} + \frac{1}{2} \frac{p_{ij}}{\rho_0 c_0} \\ P^* = \bar{p}_{ij} + \frac{1}{2} \beta \rho_0 c_0 U_{ij} \end{cases} \quad (16)$$

Here,  $(\bar{\bullet})_{ij} = [(\bullet)_i + (\bullet)_j] / 2$  represents particle-pair average,  $\bar{U}_{ij} = -\bar{\mathbf{v}}_{ij} \cdot \mathbf{e}_{ij}$  and  $U_{ij} = -\mathbf{v}_{ij} \cdot \mathbf{e}_{ij}$ , represent the particle-pair average and difference of the particle velocity along the interaction line, respectively. The low-dissipation limiter is defined as  $\beta = \min(3 \max(U_{ij}/c_0, 0), 1)$ . Additionally, it should be noted that the particle-pair pressure  $P^*$  in Eq. (16) comprises two main components: a non-dissipative term denoted by  $\bar{p}_{ij}$ , and a dissipative term, which plays the role on satisfying the second law of thermodynamics for the overall WCSPH discretization, derived from the differences between particle pairs.

Similar to Refs. [41,50] to avoid large density (and due to the invariant particle mass, the corresponding volume) errors in long-physical-time simulations, a density initialization scheme [51] is adopted to cancel the error accumulated in updating the density using a non-conservative formulation of the continuity equation Eq. (15). At the beginning of each new advection time step, the fluid density field of free-surface flows is reinitialized by

$$\rho_i = \max \left( \rho^*, \rho^0 \frac{\sum W_{ij}}{\sum W_{ij}^0} \right), \quad (17)$$

where  $\rho^*$  represents the density updated by Eq. (15) for the last several acoustic time steps and the superscript 0 denotes the initial reference value at the start of the simulation. For flows without the free surface, Eq. (17) simplifies to

$$\rho_i = \rho^0 \frac{\sum W_{ij}}{\sum W_{ij}^0} \quad (18)$$

which resets the density using the standard summation formulation [16].

## 2.3. Eulerian SPH (ESPH)

The conservation equations for weakly compressible flows in the Eulerian framework are expressed as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \quad (19)$$

where  $\mathbf{U} = (\rho, \rho \mathbf{v})$  denotes the vector of conserved variables, and  $\mathbf{F}(\mathbf{U})$  represents the corresponding fluxes. Following the methodology outlined in Refs. [52,53], the Eulerian SPH discretization of Eq. (19) can be expressed in the same anti-symmetric

or conservative form for both mass and momentum conservations as

$$\begin{cases} \frac{\partial}{\partial t} (\rho_i V_i) + 2 \sum_j (\rho \mathbf{v})_{E,ij}^* \cdot \nabla W_{ij} V_i V_j = 0 \\ \frac{\partial}{\partial t} (\rho_i \mathbf{v}_i V_i) + 2 \sum_j [(\rho \mathbf{v} \otimes \mathbf{v}) + p \mathbf{I}]_{E,ij}^* \cdot \nabla W_{ij} V_i V_j = 0. \end{cases} \quad (20)$$

Here, terms  $(\bullet)_{E,ij}^*$  denote numerical fluxes for each particle pair, determined by solutions of the Riemann problem [52]. The HLL Riemann solver [54,55] incorporating a low-dissipation limiter [5] is adopted here to solve the Riemann problem. The solutions as numerical fluxes can be written as

$$\mathbf{F}^* = \frac{1}{2} \bar{\mathbf{F}}_{ij} + \beta \left( \frac{1}{2} \frac{S_R + S_L}{S_R - S_L} \mathbf{F}_{ij} + \frac{S_R S_L}{S_R - S_L} \mathbf{U}_{ij} \right). \quad (21)$$

Here,  $S_L$  and  $S_R$  represent the wave speeds estimated in the left and right regions of the Riemann problem (see Ref. [54] for more details), respectively, satisfying the assumption of  $S_L \leq 0 \leq S_R$  and  $S_L \neq S_R$ , which is validated for weakly compressible flows. The same low-dissipation limiter  $\beta$  as in Eq. (16) is utilized to handle the dissipative component of the numerical fluxes. Again, the non-dissipative component in Eq. (21) is given by the particle-pair average of the physical fluxes.

### 3. Reverse KGC formulation

#### 3.1. Consistent correction method

Although it has been shown that the straightforward application of the KGC matrix, as outlined in Eq. (12), is able to obtain improved results for some problems [42,45,49,56] compared to the original conservative formulation in Eq. (6), it does not achieve zero- or first-order consistency for the conservative formulation. Specifically, Eq. (12) can be rewritten as

$$\nabla \psi_i = -\psi_i \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j + \sum_j \psi_{ij} \mathbf{B}_j \nabla W_{ij} V_j. \quad (22)$$

The first term on the right-hand side (RHS), mimicking Eq. (10), represents the zero-order consistency condition for incorporating the KGC, and the second term, again mimicking Eq. (9), represents the first-order consistency. However, the first term generally does not vanish even after particle relaxation driven by constant background pressure [15] due to the modification by the KGC matrix. In addition, the second term is different from the original form as the KGC matrix of neighboring particles is employed. Consequently, it does not guarantee first-order consistency either. These same issues also arise in other corrected formulations [46–48].

Therefore, we modify Eq. (22) by using the KGC matrix of particle  $i$  as

$$\nabla \psi_i = -\psi_i \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j + \sum_j \psi_{ij} \mathbf{B}_i \nabla W_{ij} V_j, \quad (23)$$

so that the second term is the same as Eq. (9) and achieves first-order consistency. If the first term also vanishes for achieving zero-order consistency, the entire formulation satisfies both consistencies at the same time. Motivated by employing constant background pressure for particle relaxation, we can consider  $\mathbf{B}_i$  and  $\mathbf{B}_j$  as “geometric stresses” dependent on the particle locations and use them to drive particle relaxation. After the particles are settled down or fully relaxed under such a KGC-driven particle relaxation, the first term vanishes. Note that Eq. (23) can be cast as the following anti-symmetric form

$$\nabla \psi_i = - \sum_j (\psi_i \mathbf{B}_j + \psi_j \mathbf{B}_i) \nabla W_{ij} V_j. \quad (24)$$

Comparing Eq. (24) with Eq. (12), one can find that the only difference is that, in the new formulation enabling both zero- and first-order consistencies, the KGC matrix is employed in reverse order with respect to particles  $i$  and  $j$ . Therefore, Eq. (24) is denoted as a reverse KGC (RKGC) formulation.

In this work, the RKGC formulation is employed in the momentum conservation equation for the Lagrangian SPH method and both the mass and momentum conservation equations for the Eulerian SPH method by replacing the original particle-pair average for the non-dissipative terms in the Riemann solutions as the form of  $\overline{(\bullet \mathbf{B})}_{ij} = [(\bullet)_i \mathbf{B}_j + (\bullet)_j \mathbf{B}_i] / 2$ . Specifically, the particle-pair average pressure in Eq. (16) and average fluxes in Eq. (21) are replaced as

$$\bar{p}_{ij} \Rightarrow \overline{(p \mathbf{B})}_{ij} = \frac{1}{2} (p_i \mathbf{B}_j + p_j \mathbf{B}_i), \quad (25)$$

and

$$\bar{\mathbf{F}}_{ij} \Rightarrow \overline{(\mathbf{F} \mathbf{B})}_{ij} = \frac{1}{2} (\mathbf{F}_i \mathbf{B}_j + \mathbf{F}_j \mathbf{B}_i), \quad (26)$$

respectively. Note that, since the dissipative terms are kept unchanged, the overall properties on satisfying the second law of thermodynamics are preserved. Also note that our present numerical experiments indicate no obvious difference in simulations when the KGC is employed in the continuity equation for the Lagrangian SPH method. Therefore, we simply keep the original discretization, and resort to further study of the KGC correction on density evolution in future work.

### 3.2. KGC-based particle relaxation and transport-velocity formulation

The original relaxation method (denoted as P relaxation or PR), as detailed in Ref. [15,39], operates by executing particle relaxation driven by a constant background pressure to achieve the zero-order consistency condition as outlined in Eq. (10). It iteratively adjusts particle positions to rectify zero-order integration errors, with the correction at each relaxation step determined by

$$\Delta \mathbf{r}_i = \alpha (\Delta x)^2 \sum_j \nabla W_{ij} V_j. \quad (27)$$

During the relaxation process, the particle volume is invariant, and no other underlying physics quantities are involved. The general effective choice of parameter  $\alpha = 0.2$  follows the selection of CFL number in the SPH method [16] to ensure time-stepping stability.

To incorporate the proposed RKGC formulation, we introduce the KGC-based relaxation (denoted as B relaxation or BR), where the particle relaxation is driven by the “geometric stress” or the KGC matrix to achieve zero-order consistency condition as suggested by the first term in Eq. (23). Similarly to P relaxation, the iterative correction on particle positions at each relaxation step is modified as

$$\Delta \mathbf{r}_i = \alpha (\Delta x)^2 \sum_j (\mathbf{B}_i + \mathbf{B}_j) \nabla W_{ij} V_j. \quad (28)$$

Note that the KGC matrix for each particle is recomputed by Eq. (8) before each iteration step. Similar to P relaxation, B relaxation also results in a uniformly distributed particle distribution, with body-fitted particles for complex geometries. In the current study, the KGC-based particle relaxation is used for generating the initial particle distribution for Eulerian SPH to combine with the proposed RKGC formulation. The threshold to stop the relaxation is when the maximum zero-order consistency error is smaller than  $10^{-5}$ . Our numerical experiments find that, generally, the average error is about one-tenth of the maximum error.

In the Lagrangian SPH method, the original transport-velocity formulation introduced by Adami et al. [40] is employed to enhance, rather than precisely achieve, zero-order consistency, and to prevent particle clustering under conditions of negative pressure. The method involves a single correction step during each time step, where the transport-velocity, denoted by  $\tilde{\mathbf{v}}$ , governs the updates of particle positions from one step to the next according to

$$\frac{d\mathbf{r}_i}{dt} = \tilde{\mathbf{v}}_i. \quad (29)$$

The transport-velocity formulation is numerically equivalent to applying a one-step correction of particle position at each advection time step according to Eq. (27) to achieve less consistency error [15].

In the current approach, the KGC-based transport-velocity formulation is proposed similarly, except that the single correction step is associated with “geometry stress” as in Eq. (28). Note that both transport-velocity formulations only slightly modify the positions of the particles without modifying the velocity or the momentum of the entire system. KGC-based transport velocity has been implemented in the Lagrangian SPH method for internal flow, as noted in Section 5, but is not applied for free-surface flow due to the complexity of handling free surface and the general practices of SPH simulations [18,45,56].

### 3.3. Variational consistency analysis

The Lagrangian variational principle and other similar variational principles have been employed to analyze the conservation properties of the fluid in SPH method [43,57,58], and the principle of virtual work (PVW) has been adopted for the in-depth analysis of the free-surface simulation under the weakly compressible hypothesis [59]. Follow the PVW analyzing in NKGC [59] and SKGC [56] to analyze RKGC, where the work done by internal forces in an equilibrium system equals the work done by external forces due to the virtual displacement field  $\delta \mathbf{r}$ . By disregarding the work caused by the motion of solid boundaries and assuming zero pressure along the free-surface boundary [56], the PVW for the entire system can be expressed as the conservative form

$$\iiint_{\Omega} \nabla \cdot (p \delta \mathbf{r}) dV = 0, \quad (30)$$

and the theoretically equivalent non-conservative form

$$\iiint_{\Omega} \nabla p \cdot \delta \mathbf{r} dV = \iiint_{\Omega} -p \nabla \cdot \delta \mathbf{r} dV, \quad (31)$$

both indicating that the energy variation induced by the virtual displacement equals zero for the entire system, ensuring energy, linear- and angular-momentum conservations [43,50].

By introducing particle approximation, the discrete form of Eq. (30) can be expressed as the summation of all discrete particles, we obtain

$$\sum_i \nabla \cdot (p \delta \mathbf{r})_i V_i = 0. \quad (32)$$

Since the RKGC formulation fulfills first-order consistency, the divergence operator  $\nabla \cdot (p \delta \mathbf{r})$  in Eq. (32) can be evaluated at each particle  $i$  as

$$\nabla \cdot (p \delta \mathbf{r})_i = \sum_j [(p \delta \mathbf{r})_i \mathbf{B}_j + (p \delta \mathbf{r})_j \mathbf{B}_i] \nabla_i W_{ij} V_j, \quad (33)$$

with second-order accuracy. By substituting Eq. (33) into the left-hand side (LHS) of Eq. (32) and exploiting the anti-symmetric form of the present conservative discretization, one has

$$\begin{aligned} \sum_i \nabla \cdot (p\delta\mathbf{r})_i V_i &= \sum_i V_i \sum_j [(p\delta\mathbf{r})_i \mathbf{B}_j + (p\delta\mathbf{r})_j \mathbf{B}_i] \nabla_i W_{ij} V_j \\ &= \sum_i \sum_j (p\delta\mathbf{r})_i \mathbf{B}_j \nabla_i W_{ij} V_i V_j \\ &+ \sum_i \sum_j (p\delta\mathbf{r})_j \mathbf{B}_i \nabla_i W_{ij} V_i V_j = 0, \end{aligned} \quad (34)$$

indicating the discrete variational consistency of the RKGC formulation. Actually, the conservative formulations, such as the standard and many other (including SKGC) formulations featuring the anti-symmetric form, are able to maintain the discrete variational consistency of Eq. (32).

Note that, in the present derivation, we obtain the discrete form directly from the original conservation form of Eq. (30) since the first-order consistency provided by RKGC leads to a sound approximation. This is different from previous analysis for a SKGC formulation [56], in which the discrete variational consistency is obtained from the non-conservative form of Eq. (31) (see their Eq. (26)). Also note that while the above-mentioned discrete variational consistencies indicate favorable numerical properties, they are not equivalent to the theoretical forms and, therefore, may not conserve system energy and linear- and angular momentum exactly due to the discretization errors. For example, while the exact linear momentum conservation can be ensured by the anti-symmetric form (ensuring discrete variational consistency), the exact angular-momentum conservation requires the resultant particle-pair forces acting along their central line [60,61], which is obviously not the case for either SKGC or RKGC formulations. Another example is that, as observed in Refs. [45,56] and will also be discussed in Section 5.3, even the theoretical variational consistency imposes a physically consistent boundary condition at free-surface, the SKGC formulations satisfying discrete variational consistency still suffers from unbounded increase (or instability) of total energy if the KGC is not switched off near the free surface. Still, another issue is that the present variational consistency analysis does not cover the density evolution equation, whose discretization very often leads to volume conservation issues as discussed in Section 2.2. Therefore, the strict prediction of the overall consistency properties, such as the validity of the second law of thermodynamics, of the present method, similar to many other SPH methods, remains an open problem.

#### 4. Error and convergence analyses

The accuracy and convergence of the SPH gradient operators in conservative form without correction (NKGC) in Eq. (6), with the original straightforward KGC (SKGC) in Eq. (12), and with the reverse KGC (RKGC) in Eq. (24), are investigated. A circle domain with the radius  $R = 1.0$  is considered, and a scalar field is initialized within the domain by the function

$$\psi(x) = e^{-10x^2}. \quad (35)$$

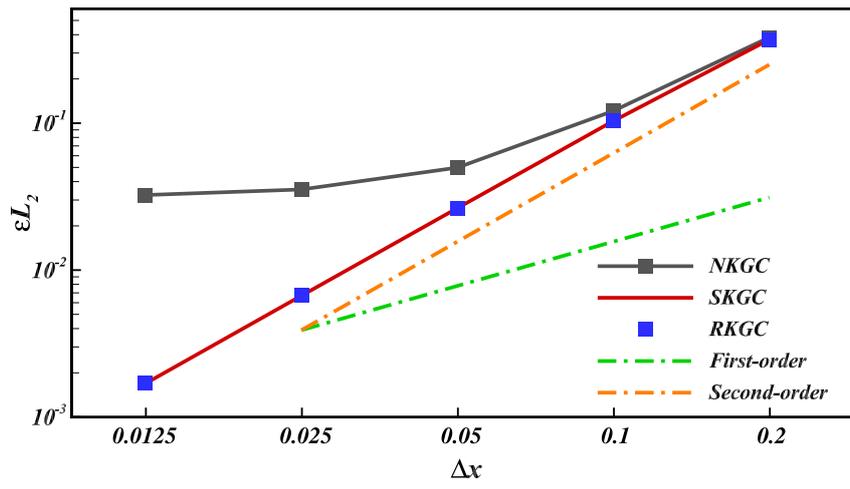
As in Ref. [14,33,42], the error is measured with  $L_2$  norm and defined as

$$\epsilon_{L_2} = \sqrt{\frac{1}{N_t} \left( \sum_i |\nabla\psi_i^{ANA} - \nabla\psi_i^{SPH}|^2 \right)}, \quad (36)$$

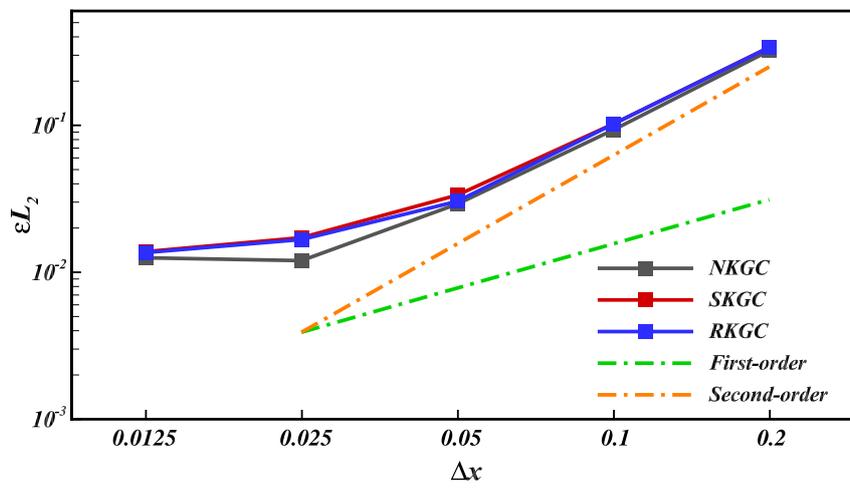
where  $\nabla\psi_i^{ANA}$  and  $\nabla\psi_i^{SPH}$  represent the analytical and numerical solutions for the gradient of the scalar field, and  $N_t$  is the total number of particles within the domain of interest (sufficiently far from the boundary). The C2 Wendland kernel is utilized to conduct tests on lattice-distributed (without body-fitted particles at the boundary) and relaxed particle distributions using both P and B relaxations. The convergence criterion for relaxation is set at a maximum zero-order consistency residual of  $10^{-5}$ . The particle spacing  $\Delta x$  ranges from 0.2 to 0.0125, while the smoothing lengths  $h$  of  $1.34\Delta x$ ,  $1.154\Delta x$ , and  $0.84\Delta x$  are used to study the convergence with decreasing  $h/\Delta x$ .

Fig. 1 presents the convergences behavior with increasing resolutions at  $h = 1.34\Delta x$ . For lattice-distributed particles, as illustrated in Fig. 1(a), both corrected formulations achieve second-order convergence compared to the typical second-order-to-saturation behavior of NKGC. For particles after P relaxation, as depicted in Fig. 1(b), all formulations exhibit the second-order-to-saturation behavior, as the integration errors are dominant at high-resolution, independently of the kernel corrections. For particles after the B relaxation, as shown in Fig. 1(c) only RKGC maintains 2nd-order convergence as expected from the analysis in Section 3.1. At high resolutions, SKGC degrades to first-order as it fails to reproduce the linear gradient, and NKGC even exhibits from error increase.

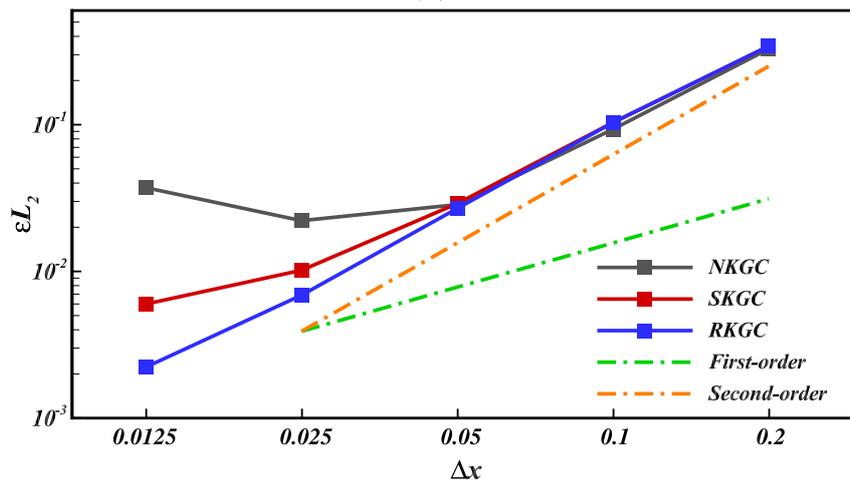
Furthermore, as depicted in Fig. 2, it is observed that for RKGC, reducing  $h/\Delta x$  does not affect the convergence rate as long as the B relaxation is applied. This is not unexpected since Eq. (23) does not explicitly depend on the smoothing length. In contrast, SKGC and NKGC suffer from serious degeneration or even increased error at high resolution, no matter whether P or B relaxation is applied. Note that data at  $\Delta x = 0.0125$  for  $h = 0.84\Delta x$  are missing for B relaxation as shown in Fig. 2(b). This is because, with the present simple relaxation stepping as given in Section 3.2, relaxation is not able to reach the zero-order residual criterion. Note that, this issue has already exhibited in Fig. 1(c) with the slight degeneration of the second-order convergence at the highest resolution due to the finite particle relaxation error threshold of  $10^{-5}$ . The difficulty of achieving sufficiently small particle relaxation error indicates an implicit limitation of the RKGC formulation. Also note that, such limitation, especially at small smoothing length, is generic and more demanding for high-order consistencies as shown latter in Section 6, as a sufficient overlap between the kernel supports of neighboring particles is required [62].



(a)

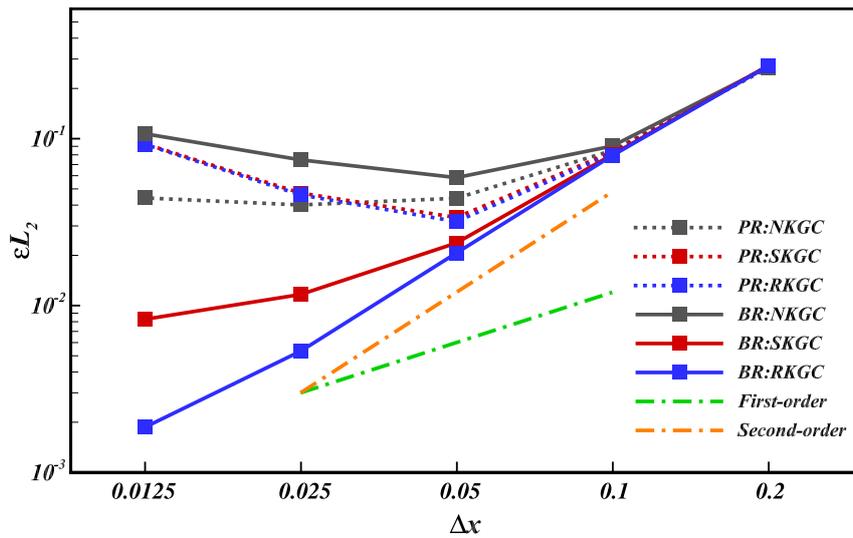


(b)

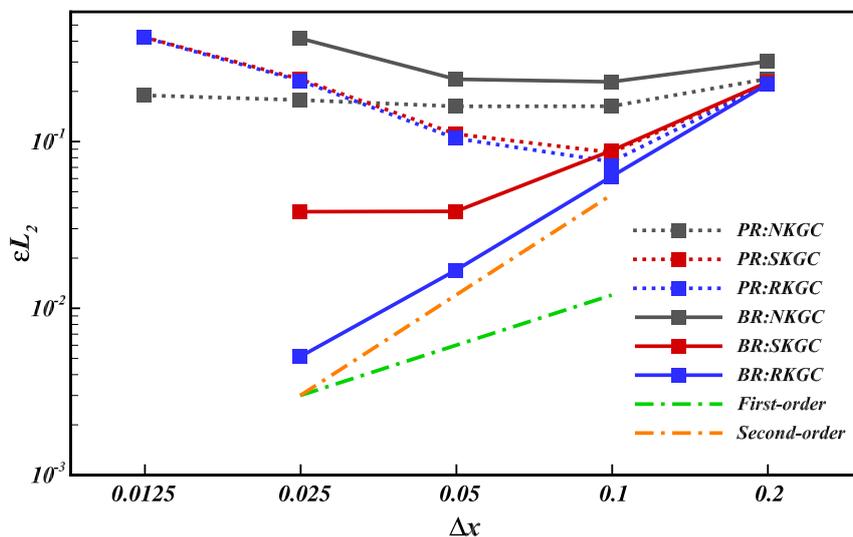


(c)

Fig. 1. Convergence study of conservative approximations of the gradient at the  $h = 1.3\Delta x$ . (a) Lattice distribution; (b) P relaxation; (c) B relaxation.



(a)



(b)

Fig. 2. Convergence study of conservative approximations of the gradient at the reduced  $h/\Delta x$  values. (a)  $h = 1.15\Delta x$ ; (b)  $h = 0.84\Delta x$ .

## 5. Numerical examples

In this section, the proposed RKGC formulation is applied to WCSPH and ESPH with fully relaxed particles for the latter. Again, the C2 Wendland kernel is utilized, with the smoothing length set to be  $h = 1.3\Delta x$  if not stated otherwise.

### 5.1. Taylor–Green vortex flow at $Re = 100$

The incompressible Navier–Stokes equation offers an analytical time-dependent solution for this periodic array of vortices in a unit square domain as

$$\begin{cases} u(x, y, t) = -U e^{bt} \cos(2\pi x) \sin(2\pi y) \\ v(x, y, t) = U e^{bt} \sin(2\pi x) \cos(2\pi y). \end{cases} \quad (37)$$

This solution serves as the initial velocity distribution at  $t = 0$  and acts as a benchmark to assess the simulation accuracy. The decay rate of the velocity field is determined by  $b = -8\pi^2/Re$ , where  $Re = \rho U L/\eta$  represents the Reynolds number derived from the fluid density  $\rho$ , the maximum initial velocity  $U$ , the domain size  $L$ , and the viscosity  $\eta$ . In the current simulations, a domain size of  $L = 1$

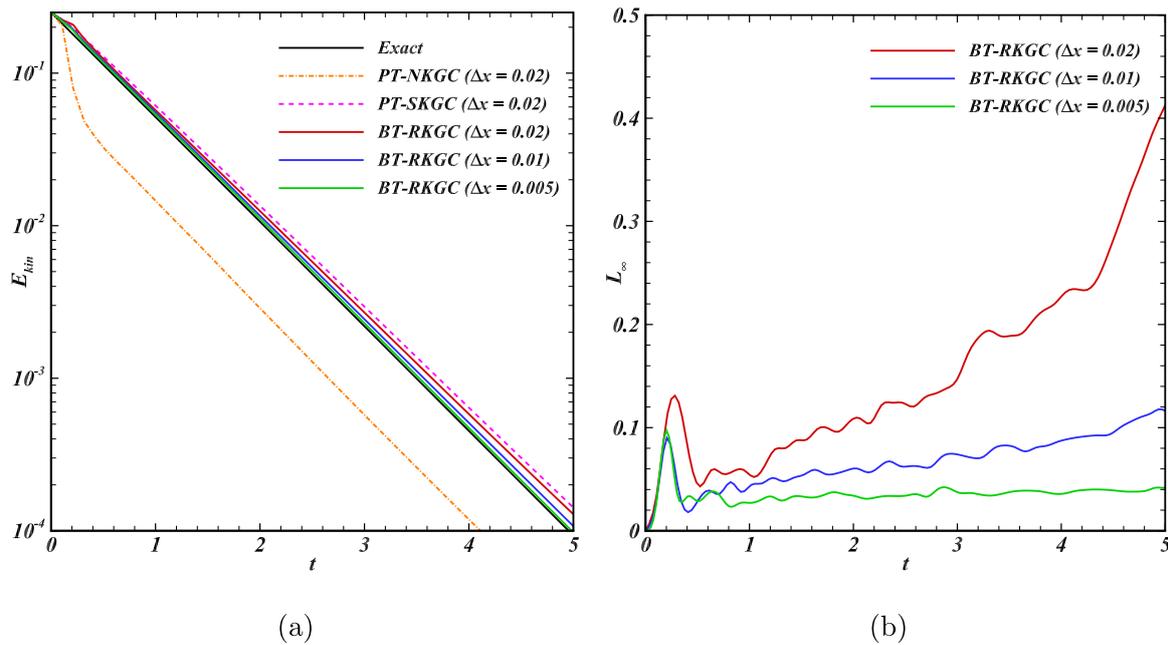


Fig. 3. Taylor–Green Vortex: WCSPH results obtained with the transport-velocity formulations. (a) Decay of kinetic energy; (b) Relative error of the maximum velocity (BT-RKGC).

is employed, with periodic boundary conditions applied in both coordinate directions. The maximum initial flow speed is set at  $U = 1$ , and the Reynolds number is  $Re = 100$ .

### 5.1.1. WCSPH results

The Taylor–Green vortex problem is first investigated using the WCSPH method, with particle spacing  $\Delta x = 0.02$  ( $50 \times 50$  particles),  $\Delta x = 0.01$  ( $100 \times 100$  particles), and  $\Delta x = 0.005$  ( $200 \times 200$  particles). The initial particle distribution follows a lattice arrangement. Fig. 3 shows the results obtained by employing the transport-velocity formulation in each advection time step. Here, PT and BT denote the original and KGC-based transport-velocity formulations, respectively, and NKGC denotes the original SPH method without KGC applied. The relative error of the maximum velocity  $L_\infty^v$  is defined as

$$L_\infty^v = \left| \frac{\max(\mathbf{v}_i(t)) - U e^{bt}}{U e^{bt}} \right|. \quad (38)$$

Fig. 3(a) shows that the corrected formulations deliver superior results compared to PT-NKGC, where no correction is employed. Specifically, BT-RKGC yields the most favorable outcomes, indicating improved accuracy. Moreover, as the resolution increases, the kinetic energy obtained by BT-RKGC converges towards the analytical solutions. Such convergence is also demonstrated in Fig. 3(b) through the relative error of the maximum velocity.

### 5.1.2. ESPH results

In the ESPH simulations, the particle distributions are initialized through relaxation, ensuring that the zero-order consistency residue diminishes to less than  $10^{-5}$  for P and B relaxations in SKGC and RKGC, respectively. The initial particle spacing is varied as  $\Delta x = 0.04$  ( $25 \times 25$  particles),  $\Delta x = 0.02$  ( $50 \times 50$  particles), and  $\Delta x = 0.01$  ( $100 \times 100$  particles) to analyze the influence of the resolution. Fig. 4 presents the decay of the kinetic energy  $E_{kin}$  and its relative error  $L_\infty^E$  defined by

$$L_\infty^E = \left| \frac{E_{kin}(t) - E_{kin}^a(t)}{E_{kin}^a(t)} \right|, \quad (39)$$

where  $E_{kin}^a$  represents the analytical kinetic energy with a decay rate of  $-16\pi^2/Re$ . Both SKGC and RKGC produce results converging to the analytical solution, while RKGC achieves lower kinetic energy error across all resolutions and exhibits improved accuracy compared to that of SKGC.

## 5.2. Lid-driven cavity at $Re = 1000$

The lid-driven cavity problem serves as a well-known and challenging test case for the SPH method. In this scenario, a wall-bounded unit square cavity is presented, with its top wall moving at a constant speed of  $U_{wall} = 1$ . For the flow at a Reynolds number of  $Re = 1000$ , we refer to the high-resolution multi-grid results of Ghia et al. [63], who utilized the finite difference method on a  $257 \times 257$  mesh.

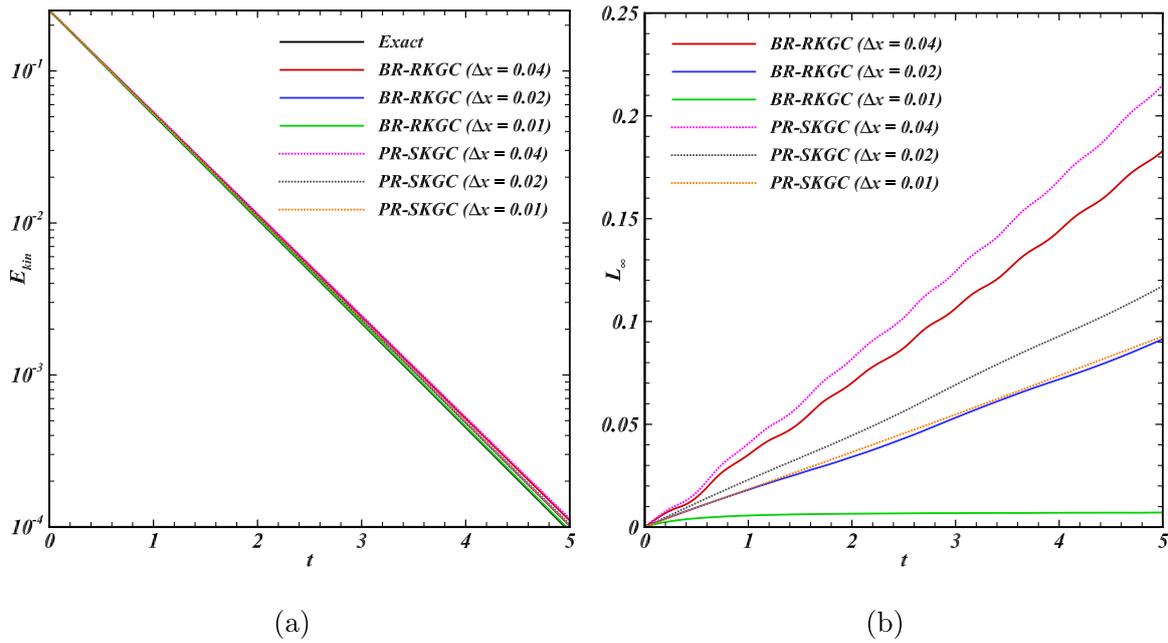


Fig. 4. Taylor–Green Vortex: ESPH results obtained with the relaxed particles. (a) Decay of kinetic energy; (b) Relative error of kinetic energy.

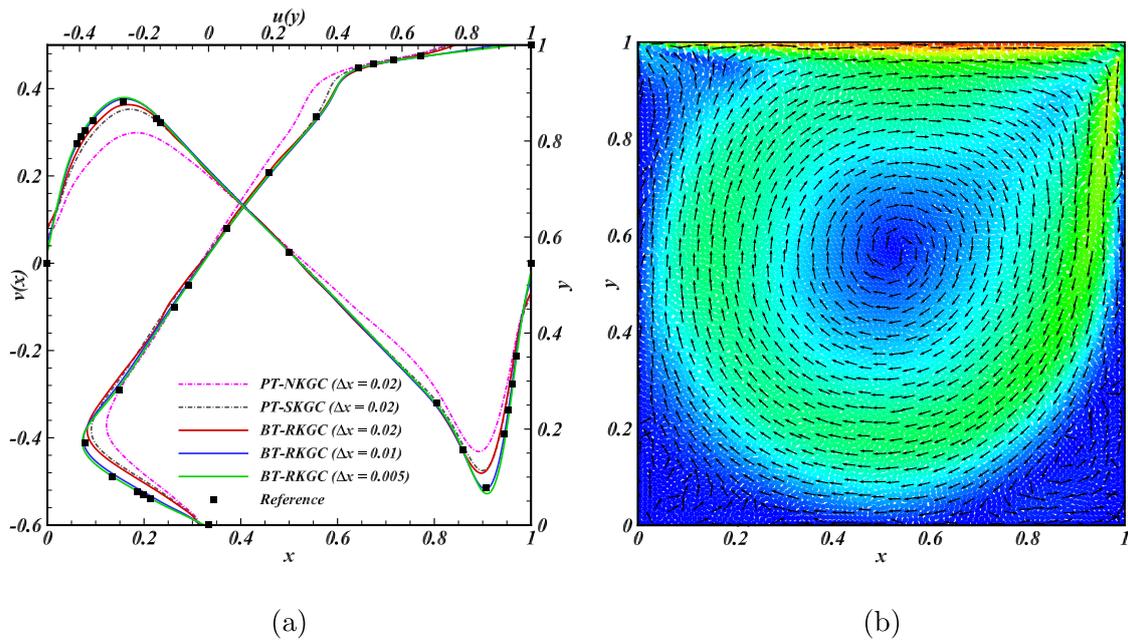


Fig. 5. Lid-driven cavity: WCSPH results obtained with the transport-velocity formulations. (a) Velocity profiles along the horizontal and vertical central lines; (b) Velocity fields (visualized by the magnitude ranging from 0 to 1) with vectors obtained by BT-RKGC at the  $\Delta x = 0.01$ .

5.2.1. WCSPH results

The results obtained by the WCSPH method are presented in Fig. 5. Comparisons of velocity profiles with the reference results, depicted in Fig. 5(a), reveals that corrected formulations yield results more closely aligned with the reference than PT-NKGC. Furthermore, BT-RKGC outperforms PT-SKGC. As resolution increases, results obtained by BT-RKGC demonstrate convergence and good agreement with the reference. The magnitude of the velocity and the velocity vectors of the flow field shown in Fig. 5(b) exhibit a smooth profile and typical vortical structures, such as those induced by the shear force of the moving wall and the single-core vortex located at the center of the cavity, consistent with findings presented in Refs. [40,63,64].

5.2.2. ESPH results

Again, the ESPH results are obtained on fully relaxed initial particle distributions at smoothing lengths of  $h = 1.3\Delta x$  and  $h = 1.15\Delta x$ , as illustrated in Fig. 6. It is observed that the BR-RKGC produces results closer to the reference than the PR-SKGC

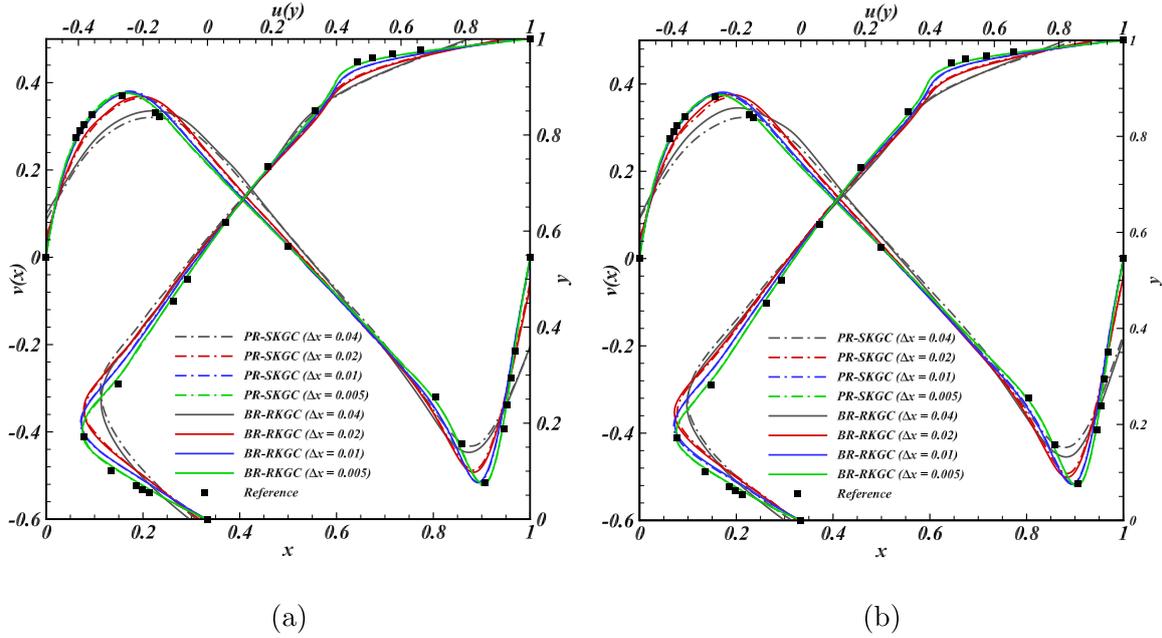


Fig. 6. Lid-driven cavity: ESPH method results for velocity profiles along the horizontal and vertical central lines at different smoothing lengths. (a)  $h = 1.34\Delta x$ ; (b)  $h = 1.15\Delta x$ .

at the low resolutions, i.e.,  $\Delta x = 0.04$  and  $\Delta x = 0.02$ , indicating less integration errors. With the resolution increased, both methods could yield converged results, and the error difference is small due to the sufficient smoothing length.

Fig. 7 presents the obtained velocity fields with vectors when the smoothing length is decrease to  $h = 0.8\Delta x$ . As discussed in Section 4, the RKGC formulation combined with the B relaxation achieves the first-order consistency, not explicitly relying on the smoothing length, while the SKGC one suffers serious degeneration. Therefore, as presented in Fig. 7(a), the BR-RKGC is still able to generate the smooth velocity distribution and captures the key flow characteristics, aligning with the reference results [63,65] and the one displayed in Fig. 5(b) obtained by the WSPH method. Moreover, the secondary vortices in the lower and upper corners are still well-identified. However, the PR-SKGC, as presented in Fig. 7(b), fails to yield a reasonable smooth velocity distribution, with velocity oscillation and noise, especially at the upper-left corner vortex and around the single-core vortex. The velocity vectors at the lower corners also failed to capture the secondary vortices. The proposed RKGC formulation and B relaxation can still achieve improved accuracy and good convergence, even with a reduced  $h/\Delta x$  value. Note that in practical viscous flow simulations, while the RKGC aims to achieve first-order consistency for gradient and divergence operators, the numerical results do not necessarily always maintain it because the Laplacian operator in SPH approximations does not precisely satisfy first-order consistency yet, and the adoption of the Riemann solver may also influence the overall consistency as the RKGC is only applied on the non-dissipative terms.

### 5.3. Standing wave

The standing wave problem serves as a typical benchmark for evaluating the accuracy of the SPH method in addressing free-surface problems. In this section, a two-dimensional standing wave problem is investigated using the WSPH method. The initial configuration, as depicted in Fig. 8, defines the initial free surface according to

$$\eta_0 = A \cos(k(x + \lambda)/2). \quad (40)$$

Here, the parameters are set as follows: the wave amplitude  $A = 0.1H$ , with an average water depth of  $H = 1.0$ ; the wave number  $k = 2\pi/\lambda$ , and the wavelength  $\lambda = 2.0$ . The initial velocity of the particles is set to be zero. The study evaluates the free-surface elevation at the central position, and compare it against the second-order analytical solution proposed by Wu et al. [66]. In order to maintain numerical stability, the KGC matrix near the free surface is weighted by the identity matrix ( $WKGC^1$ ), as suggested in Ref. [45], with the constant parameter  $\alpha = 0.5$  for all free-surface flow problems in the current study.

The snapshots of the density distribution obtained from Eq. (17) and velocity divergence distribution from the continue equation are shown in Fig. 9, respectively, indicating good volume conservation property of the present method. In addition, the pressure contours with  $\Delta x = 0.005$  at two typical instants are illustrated in Fig. 10, showing the robust free-surface profile and smooth pressure fields generated by the RKGC formulation. Assuming a rigid body (particle) system, in this problem, the total mechanical energy is defined as the sum of potential energy and kinetic energy [67–69], and the mechanical energy decay is defined as

$$\Delta E = \frac{E_{kin} + E_{pot} - E_{pot}^0}{E_{pot}^0 - E_{pot}^\infty}, \quad (41)$$

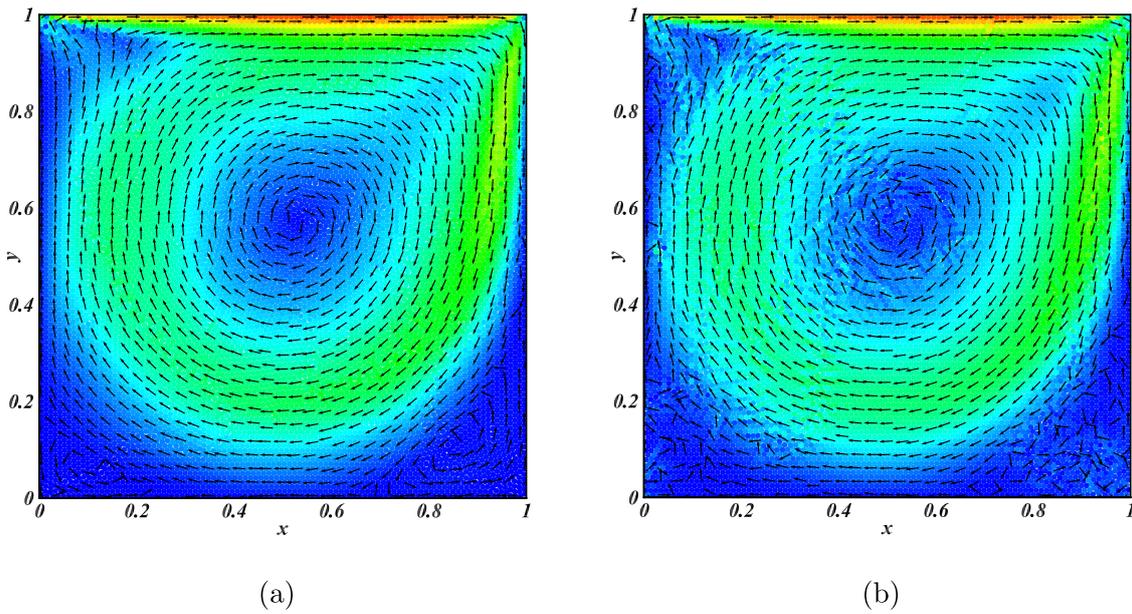


Fig. 7. Lid-driven cavity: ESPH method results for velocity fields (visualized by the magnitude ranging from 0 to 1) with vectors at  $h = 0.8\Delta x$  and  $\Delta x = 0.01$ . (a) BR-RKGC; (b) PR-SKGC.

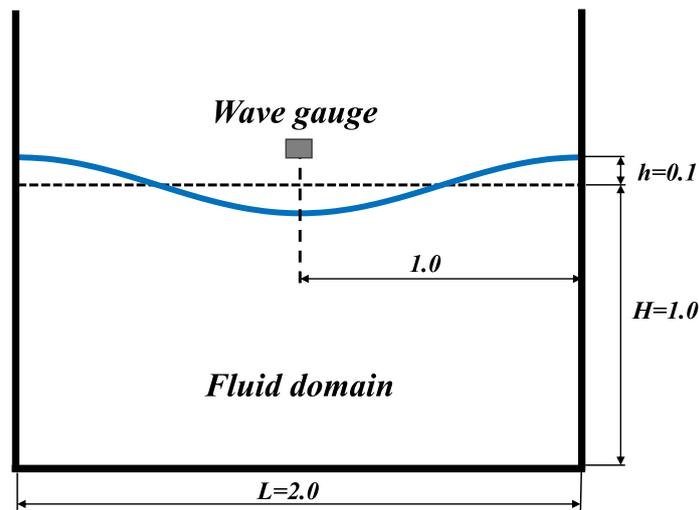


Fig. 8. Standing wave: Initial configuration of the simulation.

where  $E_{pot}^0$  and  $E_{pot}^\infty$  are the initial and the final potential (also total) energies of the system, respectively. Note that such an assumption neglects the pressure and the deformation (or elastic) contributions to the mechanical energy, and may lead to noticeable discrepancies, as shown in the following discussion, to that of the weakly compressible fluid simulation here. Fig. 11 illustrates the decay of the normalized mechanical energy across different formulations and compares them with the analytical and reference results. It is observed that the RKGC formulation is able to preserve the energy very well, suggesting very small numerical dissipation. Note that the slight but noticeable decay and recovering of the mechanical energy at early stage may attribute to the suddenly change of pressure and elastic energies which are neglected in Eq. (41) given by the rigid-body system. However, NKGC exhibits rapid energy decay, even when the smoothing length is increased to  $h = 2.0\Delta x$ , as shown by Khayyer et al. [68]. It is also noted that the SKGC formulation leads to an increase in the energy, consistent with findings from Ref. [45,56]. Therefore, extra weight with the identity matrix (as WKGC<sup>2</sup> in Ref. [45]) is added to decrease the contribution of the SKGC formulation to eliminate the artifact but, as illustrated in Fig. 11, it still shows considerable energy loss.

Fig. 12 illustrates the wave heights across different methods as well as the convergence analysis of the RKGC formulation. The comparisons of the wave height depicted in Fig. 12(a) indicate that while the results obtained with both SKGC and RKGC achieve notable improvement compared to NKGC, RKGC further improves accuracy considerably and generates results closer to the analytical solution. As displayed in Fig. 12(b), RKGC demonstrates good convergence with increasing resolution. The observed frequency discrepancy from the analytical solution at a late time is not unexpected since only a 2nd-order approximation is employed

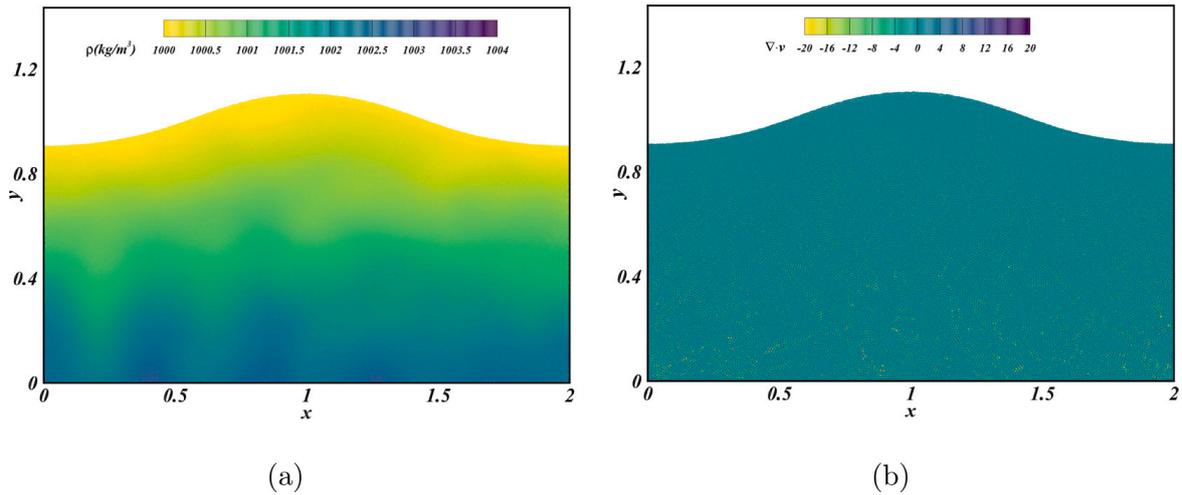


Fig. 9. Standing wave: Snapshots of (a) the density field and (b) the velocity divergence field.

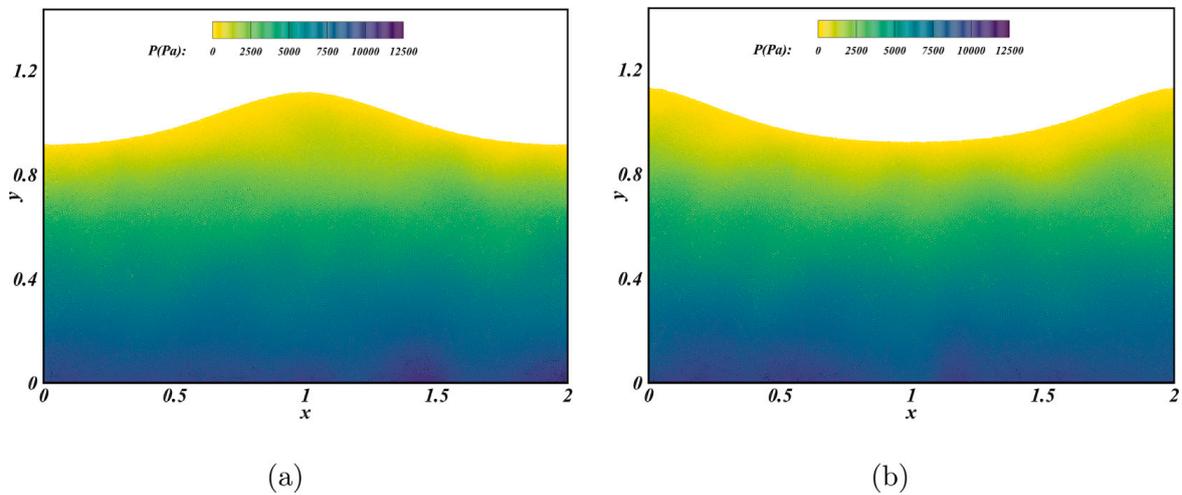


Fig. 10. Standing wave: Snapshots of the free-surface profile and the pressure contour obtained by the RKGC ( $\Delta x = 0.005$ ). (a)  $t = 23.5$  s; (b)  $t = 25.2$  s.

to obtain the analytical solution [66]. Note that similar discrepancies have also been exhibited in previous numerical results at high resolutions (see Figs. 15 and 6 in Refs. [45,68], respectively.) Regarding the computational costs, RKGC and SKGC exhibit about the same costs as given in Ref. [45]. With the adoption of dual-time stepping techniques [51], there is an additional 10% computational cost for calculating KGC matrices.

#### 5.4. Oscillating drop

The two-dimensional oscillating drop was also investigated to evaluate the energy conservation properties of the proposed method. This problem, as outlined in Ref. [70] and depicted in Fig. 13, involves a drop with a radius of  $R = 1$  immersed in an assumed inviscid fluid. This drop experiences a central conservative force  $F = -\Omega^2 R$  and is initialized with a velocity profile defined by

$$\begin{cases} u_0 = A_0 x \\ v_0 = -A_0 y, \end{cases} \quad (42)$$

where  $A_0 = 1.0$  and  $A_0/\Omega = 1.0$ . The analytical solution reported in Ref. [71] is referenced for quantitative comparison and validation.

The pressure contour with  $\Delta x = 0.005$  at two different instants are presented in Fig. 14, indicating the robust free-surface profile and smooth pressure fields obtained by the RKGC formulation. Furthermore, Fig. 15 illustrates pressure observations at the center of the domain obtained by the RKGC for different resolutions. It shows that the RKGC has good numerical stability and accuracy. As the resolutions increase, they converge and indicate a good convergence property.

Fig. 16 presents the time evolution of the decay of normalized mechanical energy across different formulations and compares them with the analytical and reference results. Aligning with the observation in Fig. 11, the RKGC formulation is able to preserve

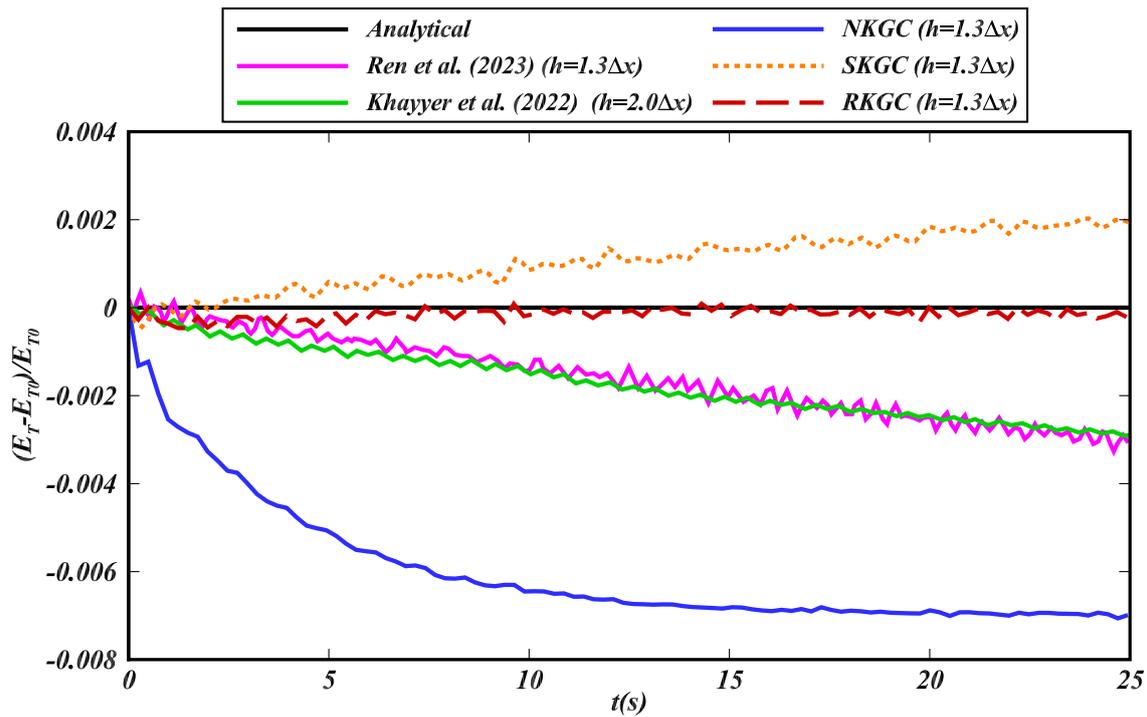


Fig. 11. Standing wave: Time evolution of the decay of the normalized mechanical energy obtained by different formulations ( $\Delta x = 0.01$ ).

the energy quite well, even at a low resolution, and the energy conservation properties are improved with the resolution increase. It indicates there is no evident energy decay at the resolution of  $\Delta x = 0.005$ . However, at this high resolution, NKGC still exhibits a considerable energy decay rate. While the corrected formulations introduced by Ren et al. [45] and Huang et al. [49] achieve notable reduction of the energy decay, they still suffer more energy loss than that obtained by RKGC at a lower resolution. The separate comparisons of kinetic and potential energies are depicted in Fig. 17. The corrected formulations all show improved agreement with the analytical solution compared to the NKGC for both kinetic and potential energies. Moreover, RKGC exhibits a closer agreement with the analytical solution than those reported in Ref. [45] where a corrected formulation is also adopted. The separate kinetic and potential energies obtained by RKGC at different resolutions are presented in Fig. 18 as further evidence. The figure demonstrates that RKGC maintains good energy conservation across different resolutions. As the resolution increases, the results converge, and no evident energy decay is observed at high resolution, agreeing with the observation in Fig. 16, and the analytical solution shows a slight energy increase over time, indicating some error against the exact energy conservation.

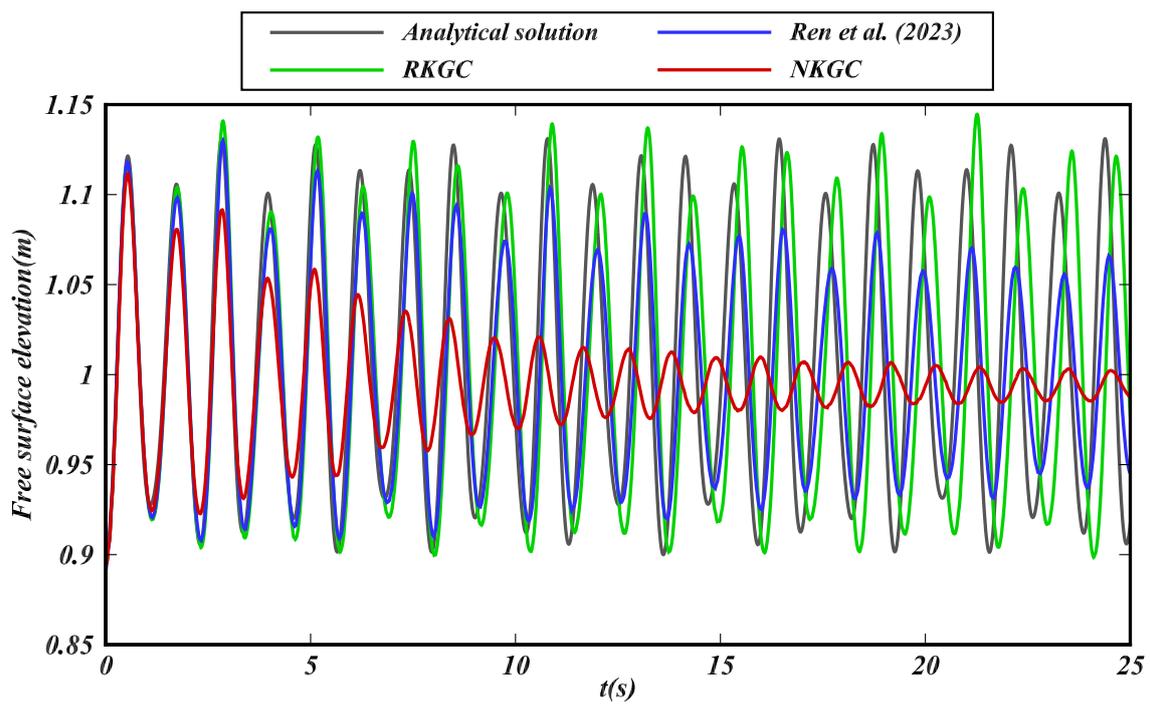
### 5.5. Progressive wave propagation

The two-dimensional numerical wave tank (NWT) has been used to simulate the progressive gravity wave. Following the configuration in Ref. [47], as shown in Fig. 19, the tank has a flat region of length  $l_f = 50$  m with a still water depth of  $d = 1$  m, and it terminates with a 10 m long sloping section, where the slope is 10%, to reduce the wave reflection from the end. The simulated waves are generated using a piston-type wavemaker, producing waves with a height  $H = 0.08$  m (from peak to trough) and a wavelength of  $\lambda = 1.5$  m. The particle spacing is set at  $\Delta x = 1/64$  m and  $\Delta x = 1/128$  m, respectively, and the simulation is run up to a time that is necessary for the wave train to reach the end of the flat region.

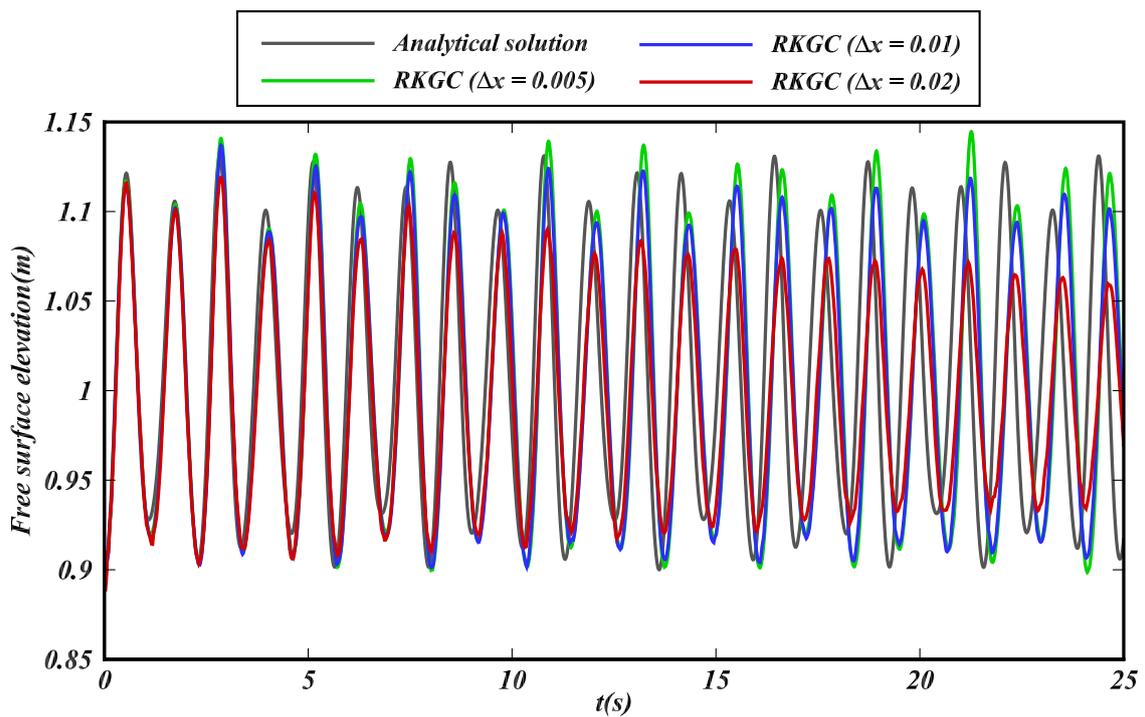
Fig. 20 illustrates the water surface elevation within the NWT relative to the mean water level, and a distance corresponding to 30 wavelengths is presented. Compared to the result without correction (see Fig. 11 in Ref. [47]), the low-resolution results obtained by the RKGC formulation already show great alleviation of the decay, and the high-resolution results do not show apparent decay, aligning with the theoretical amplitude envelope, indicating good energy conservation. Note that, while the present results are in good agreement with those reported in Ref. [47] (their Fig. 11) obtained by the CCSPH (Conservative Corrected SPH) method at the same resolution, the present computational cost is expected to be considerably lower, as CCSPH method requires the inversion of correction matrix for each particle pair at every time step and an elaborative free-surface treatment. Moreover, some irregularities in the surface elevation, similar to those in Ref. [47], have been noted. These irregularities may be attributed to spurious waves in the tank, which are related to nonlinearities generated by the wavemaker and startup-related seiching phenomena [72].

### 5.6. Dam-break flow

The dam-break flow, extensively explored both experimentally [73–76] and numerically [5,51,77,78], is a challenging benchmark to validate the SPH method. Fig. 21 presents the initial configuration for the simulation, aligning with the experimental setup



(a)



(b)

Fig. 12. Standing wave: Time evolution of the free-surface elevation at the center of the tank. (a) Comparison across different formulations ( $\Delta x = 0.005$ ); (b) Convergence study of the RKGC formulation.

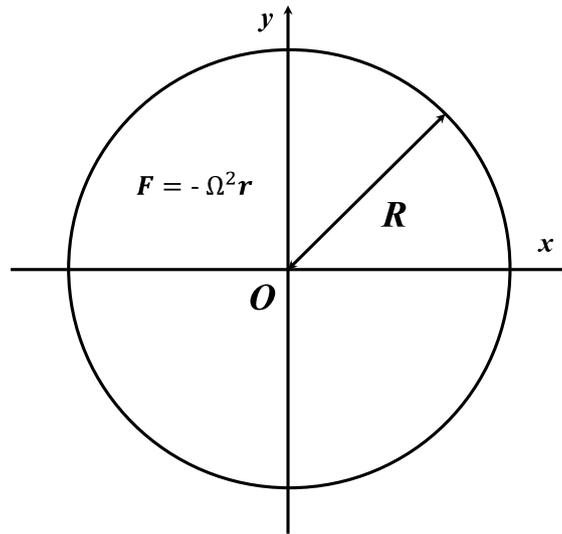


Fig. 13. Oscillating drop: Schematic illustration of the benchmark test.

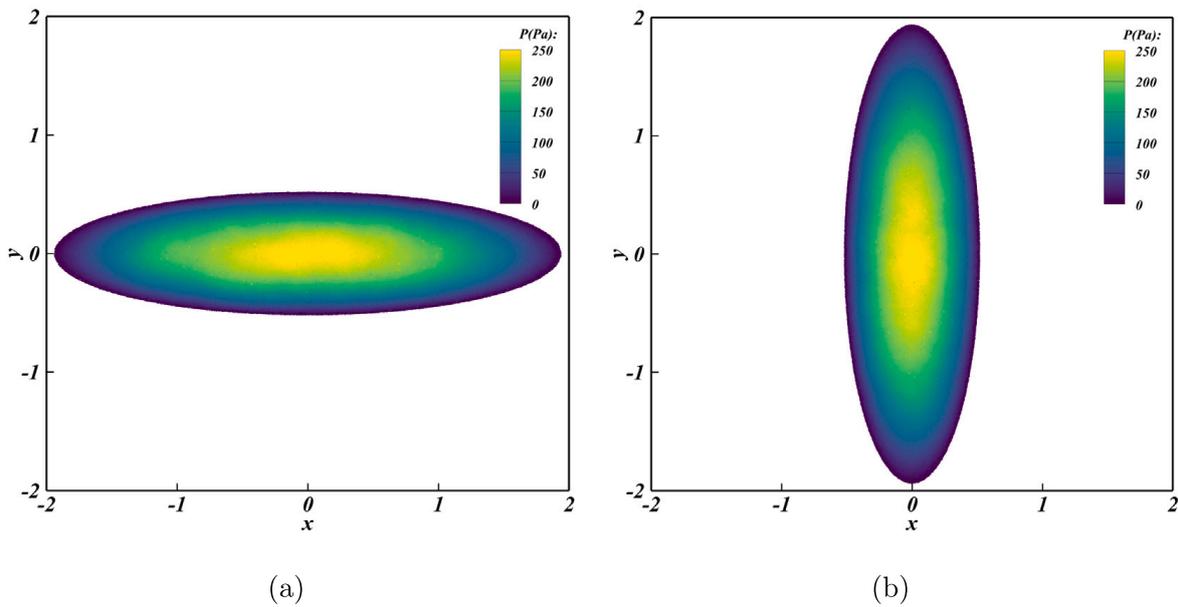


Fig. 14. Oscillation drop: Snapshots of the free-surface profile and the pressure contour obtained by the RKGC ( $\Delta x = 0.005$ ). (a)  $t = 20.5$  s; (b)  $t = 22.9$  s.

outlined by Lobovsky et al. [76]. Three measurement points (W1, W2, and W3) are assigned to record the free surface height, and three probes (P1, P2, and P3) are employed to capture the pressure signals. We consider an inviscid flow with a density of  $\rho_0 = 1000 \text{ kg/m}^3$  and a gravitational constant of  $g = 9.8 \text{ m/s}^2$ . According to the shallow water theory [79], the maximum velocity is estimated as  $2\sqrt{gH}$  to determine the speed of sound, where  $H$  represents the initial water depth.

Fig. 22 shows several typical snapshots of the time evolution of the free surface obtained by the RKGC formulation. The obtained results demonstrate smooth pressure distributions and robust free-surface profiles and align well with experimental observations [76] and previously reported simulation results [5,45,51]. RKGC could appropriately capture key flow characteristics, including high roll-up along the downstream wall, a prominently reflected jet, and free surface disruption caused by the arrival of the secondary wave. Fig. 23 shows the predicted propagation of the surge-wave front, along with comparisons to data measured in various experiments [73,74,76] and the analytical solution derived from the shallow-water equation [79]. It is observed that the present results show good convergence and agree well with the experiments before  $t\sqrt{gH} < 1$ , aligning closely with the analytical solution afterwards, but overestimate the front speed obtained from the experiments, which was also observed in other simulations [5,77,80].

The comparisons of the water levels recorded at W1, W2, and W3 with experimental observations obtained from Ref. [76] are presented in Fig. 24 and Table 1 provides the corresponding root mean square error (RMSE). The wave height exhibits good agreement with the experimental data. Nonetheless, we observe discrepancies in higher run-up waves (W1) and marginally faster

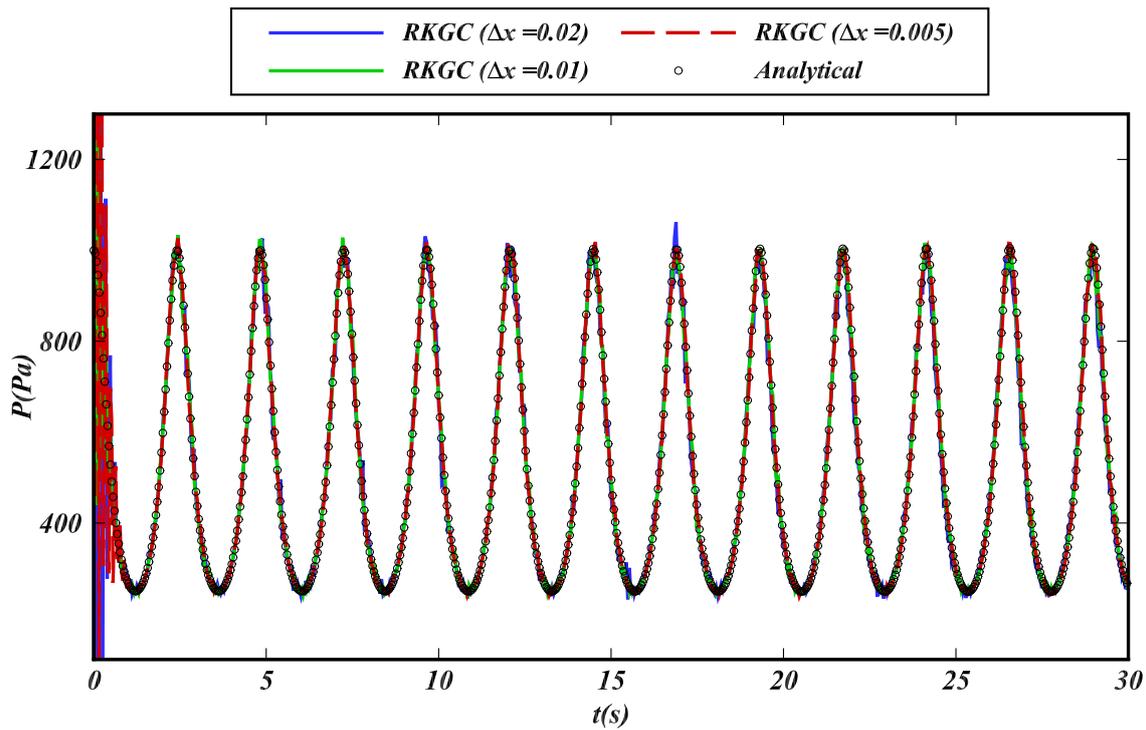


Fig. 15. Oscillating drop: Time history of the pressure at the drop center obtained by the RKGC with different particle resolutions.

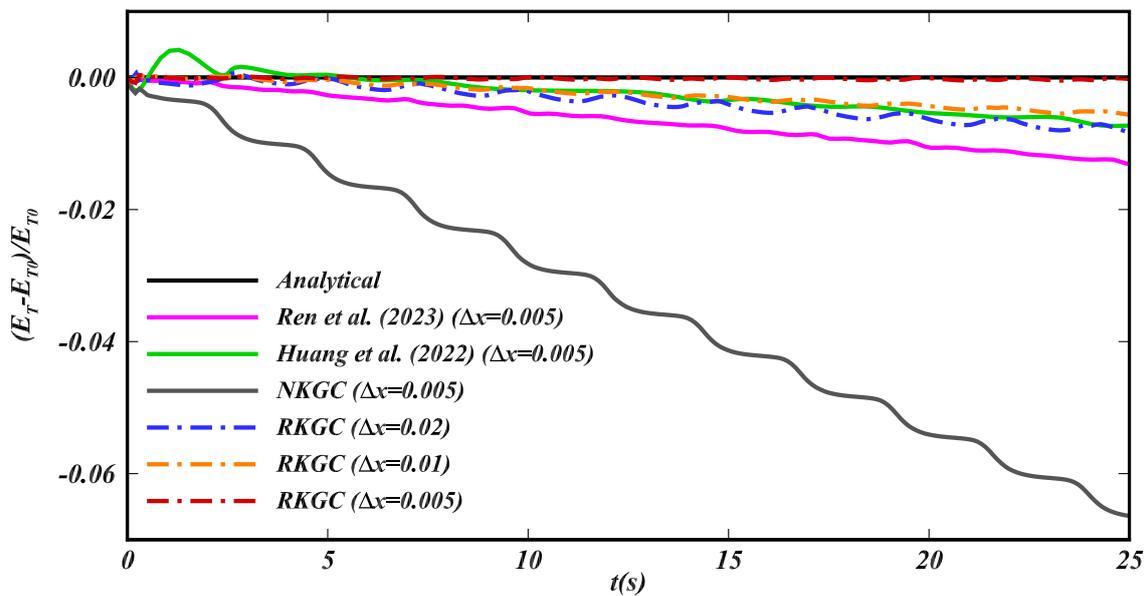
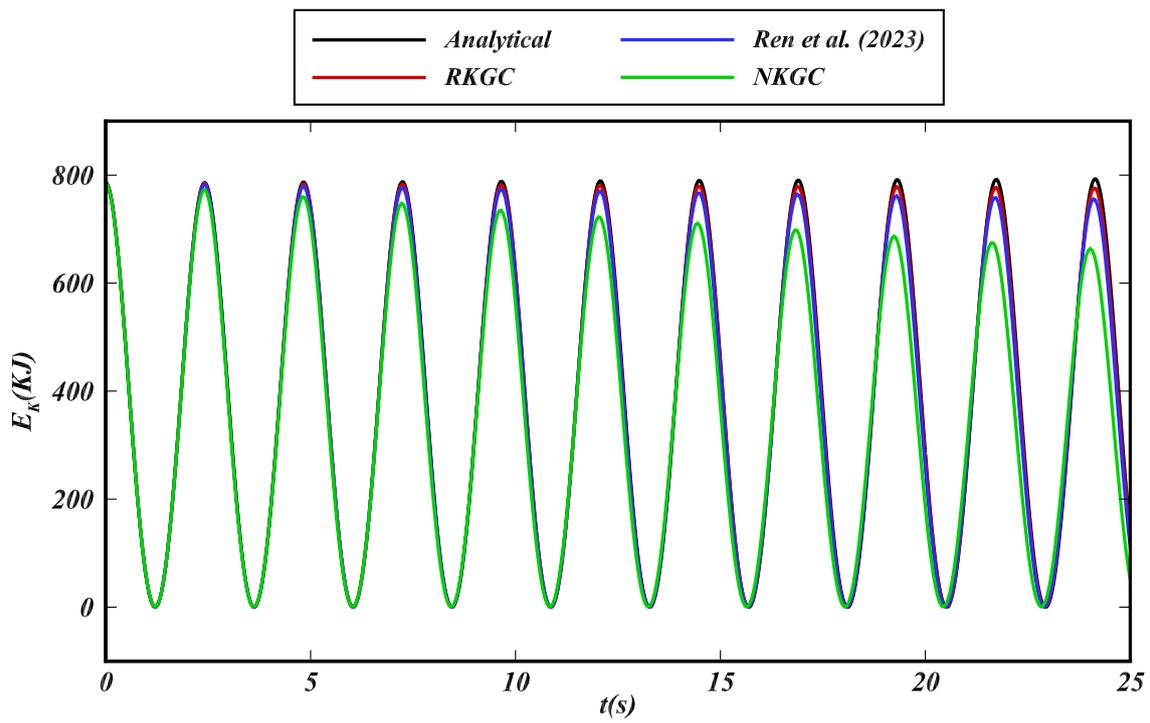


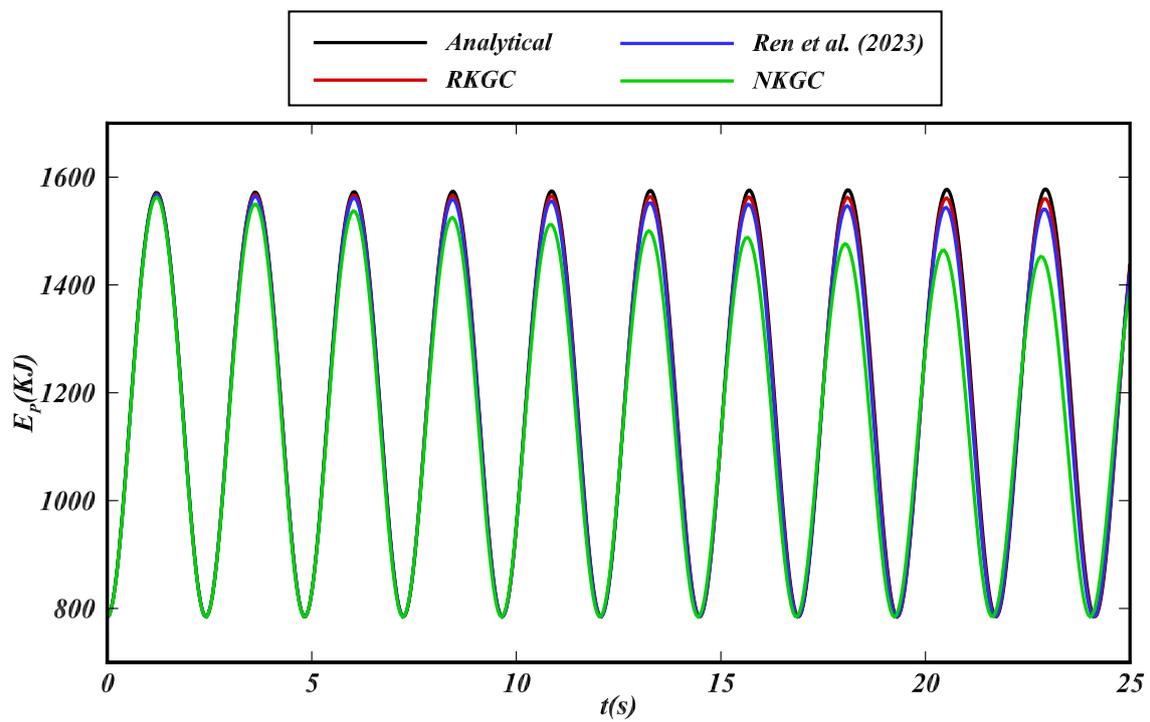
Fig. 16. Oscillating drop: Time evolution of the decay of the normalized mechanical energy obtained by different formulations.

wavefront (W2 and W3) in the current results. Similar observations were also reported in previous numerical studies [5,45,77], potentially attributed to the adoption of inviscid flow in the current study, leading to violent wave breaking up and splashing. In addition, the water level obtained at later stages shows some discrepancies due to particle splashing and accompanying the second wave to the wave breaking and re-entry, and high-resolution recognizes the small splashing structure, possibly leading to increased discrepancies, which has also been observed in Refs. [5,45,51].

The history of pressure signals recorded at P1, P2, and P3 is presented in Fig. 25. The current results have good agreement with the experimental observations [76], except for observed pressure fluctuations in the current study resulting from the weakly compressible assumption, which tend to decrease with increasing spatial resolutions. Discrepancies in pressure magnitudes at P2 and P3 were also reported in other studies [51,78] where different WCSPH methods were employed. The overestimated pressure peak at P2 by the current method is potentially due to the weakly compressible model. In addition, the air cushion effect in the

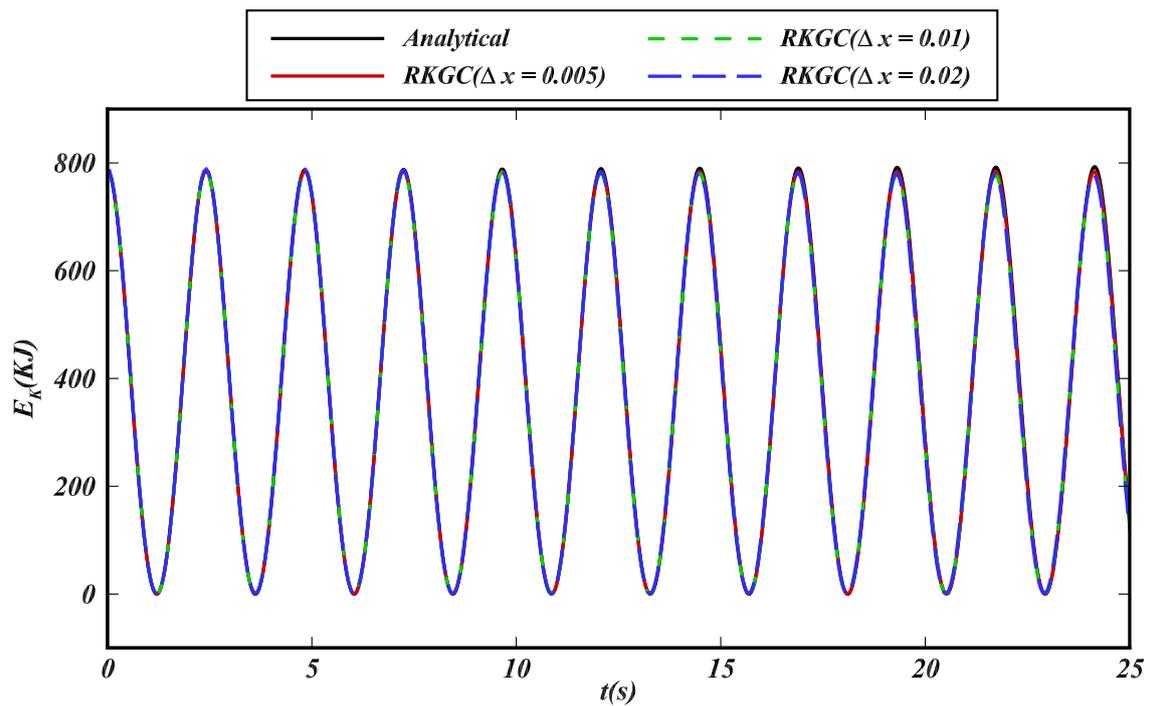


(a)

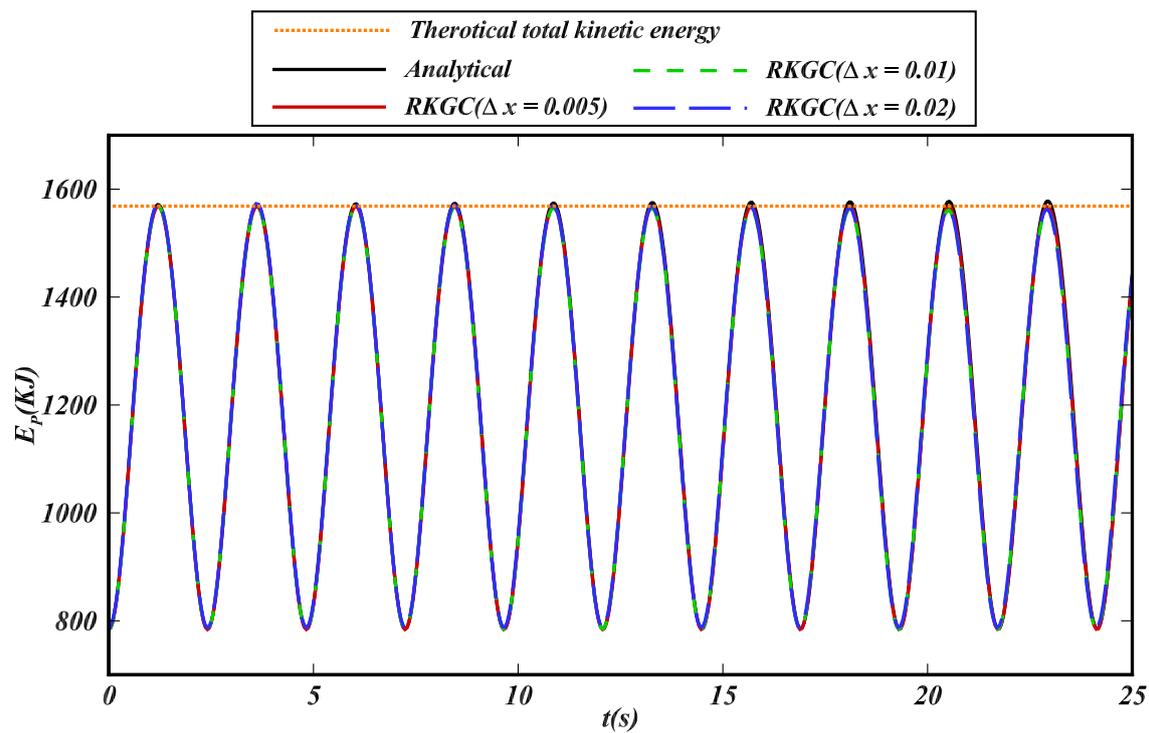


(b)

Fig. 17. Oscillating drop: Time history of the energy obtained by different formulations ( $\Delta x = 0.01$ ). (a) Kinetic energy; (b) Potential energy.



(a)



(b)

Fig. 18. Oscillating drop: Time history of the energy obtained by RKGC at different resolutions. (a) Kinetic energy; (b) Potential energy.

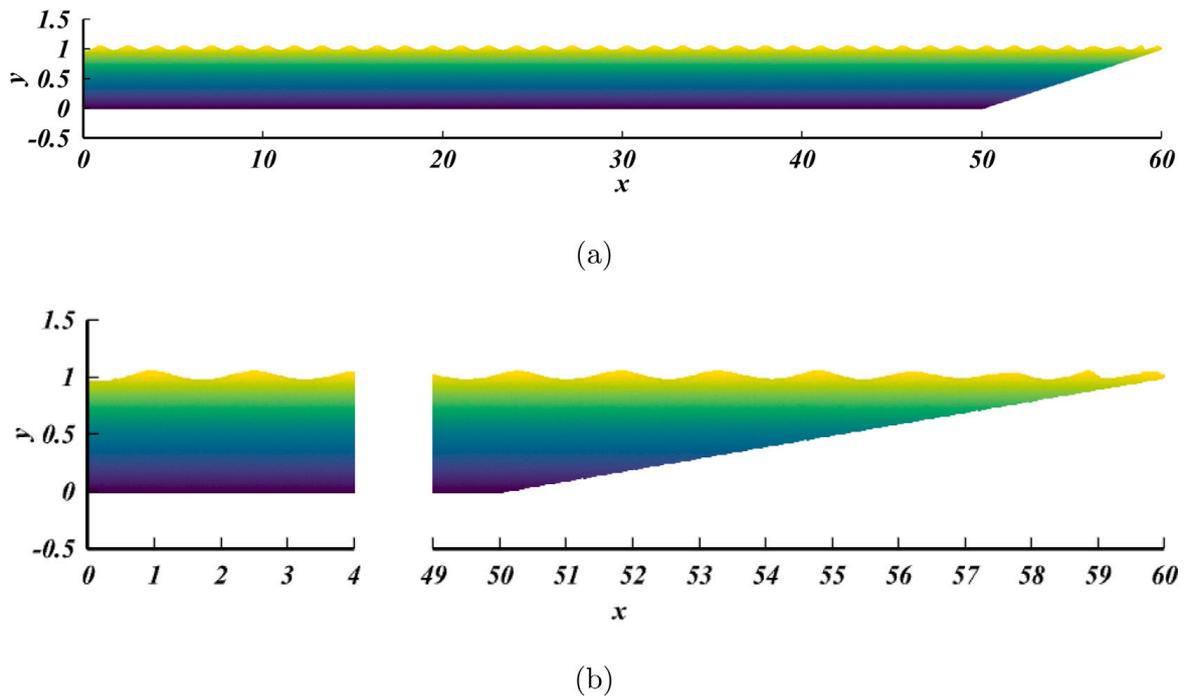


Fig. 19. Progressive wave propagation: Configuration of the NWT where waves with a height of  $H = 0.08$  m and a wavelength of  $\lambda = 1.5$  m are generated in a depth of 1 m. (a) Overall view of the NWT; (b) Zoom-in view at the beginning and the end.

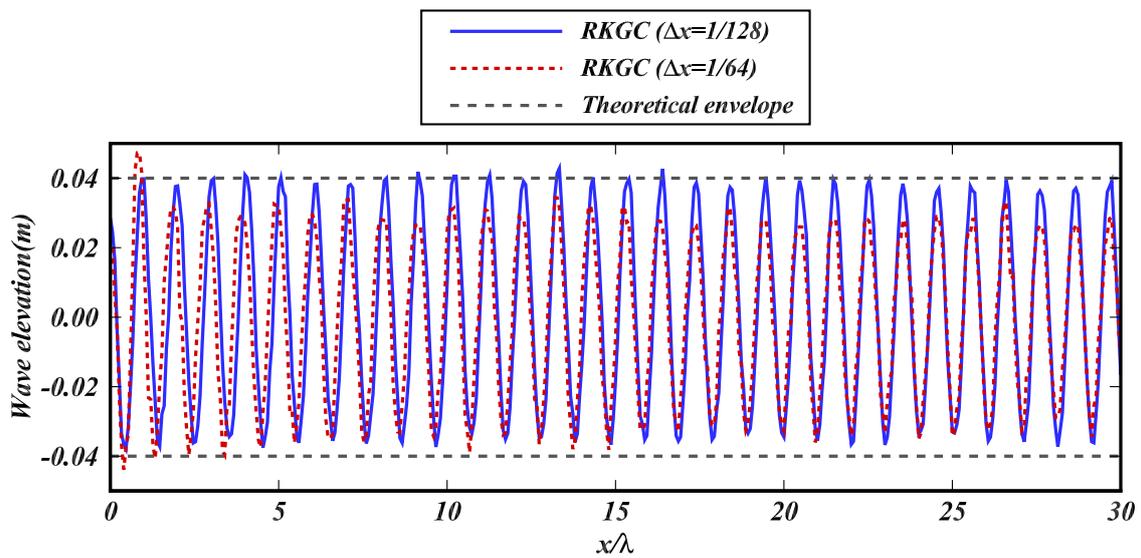


Fig. 20. Progressive wave propagation: Water surface elevation in the NWT with respect to the mean water level.

Table 1

Dam-break flow: RMSE of numerical results against experimental observations for water levels.

	$H/\Delta x = 40$	$H/\Delta x = 60$	$H/\Delta x = 80$
W1	0.05623	0.04445	0.06169
W2	0.14055	0.12691	0.14346
W3	0.21604	0.15134	0.14590

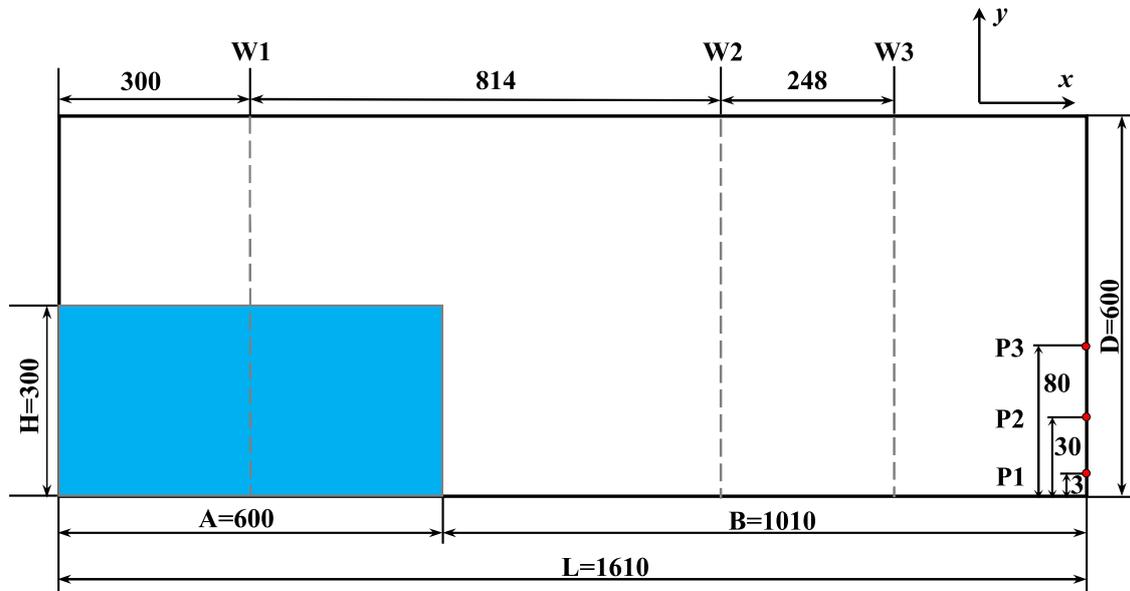


Fig. 21. Dam-break flow: Initial configuration of the simulation.

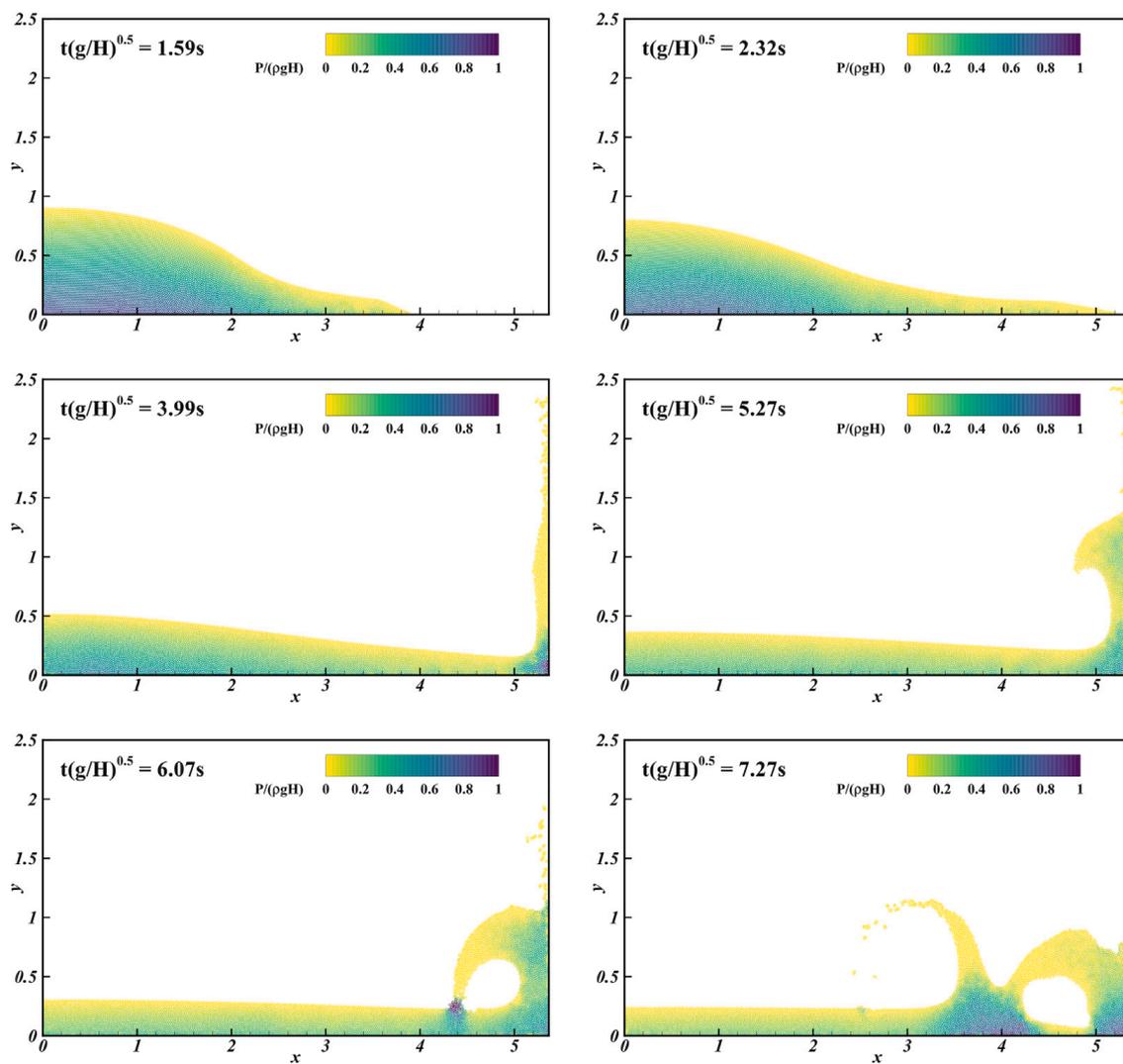


Fig. 22. Dam-break flow: Snapshots of particles and pressure distributions during the time evolution with  $H/\Delta x = 60$ .

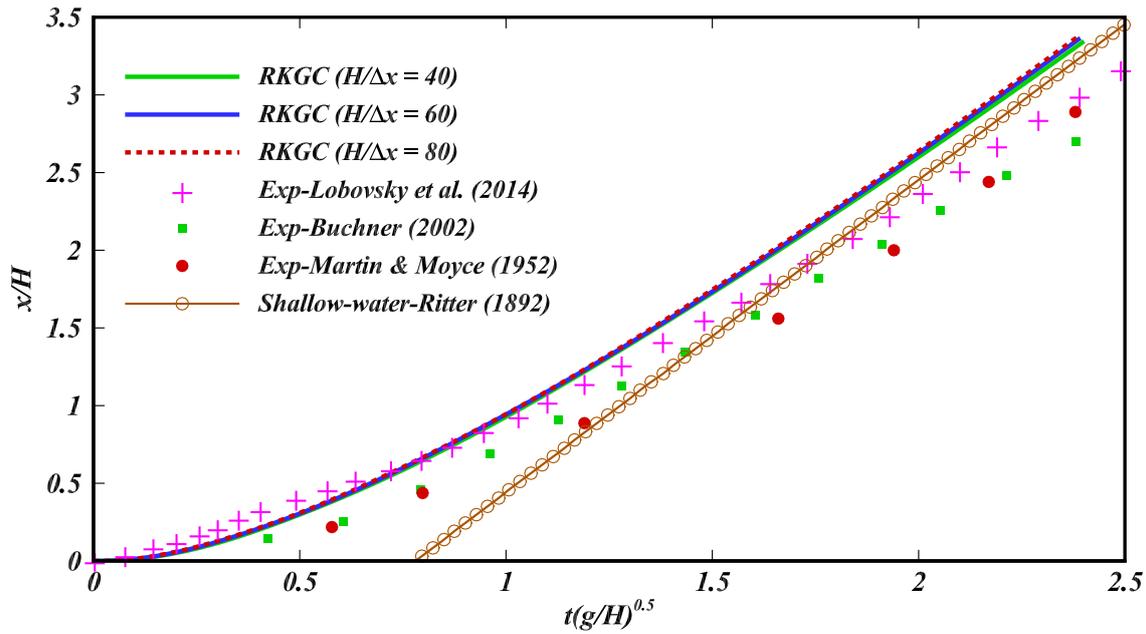


Fig. 23. Dam-break flow: Time evolution of the surge-wave front.

Table 2

OWSC: Pressure sensor positions on the front flap along the z-axis from the device's center, with  $y = 0$  representing the mean water level.

No.	y-axis (m)	z-axis (m)	No.	y-axis (m)	z-axis (m)
P01	-0.046	0.468	P09	-0.117	0.156
P03	0.050	0.364	P11	0.025	0.052
P05	-0.300	0.364	P13	-0.239	0.052

experiment may also decrease its pressure peak [76]. The occurrence time of the pressure observation at P3 aligns with  $t\sqrt{g/H} \approx 2.7$  very well, though the peak value is underestimated.

The dissipation of total mechanical energy is given as [81]

$$\Delta E = \frac{E - E_0}{E_0 - E_\infty}, \tag{43}$$

where  $E$  the total mechanical energy,  $E_0$  is the initial mechanical energy, and  $E_\infty$  is the mechanical energy after reaching the equilibrium state. Fig. 26 displays the evolution of the mechanical energy of RKGC, and compares it with SKGC and different references. Fig. 26(a) shows that with increased resolutions, the numerical dissipation decreases rapidly. It is observed in Fig. 26(b) that RKGC has lower energy decay compared to other numerical results [5,81,82], where no correction was employed. SKGC results in energy increasing before the splashing ( $t\sqrt{gH} < 6$ ), and involving the extra weighting of the KGC employed by Ren et al. [45] could alleviate this issue, but it still leads to an energy increase before the roll-up wave ( $t\sqrt{gH} < 2.3$ ). However, RKGC could maintain the energy before the splashing and leads to a slightly lower energy decay rate afterwards compared to SKGC and the results in Ref. [45], except for the energy-increasing artifacts observed for the latter.

### 5.7. Three-dimensional oscillating wave surge converter (OWSC)

As a forefront wave energy converter, the oscillating wave surge converter (OWSC) has shown remarkable energy absorption capabilities and hydrodynamic performance, and it has been widely studied with numerical and experimental methods [83–86]. In this section, the three-dimensional OWSC is investigated with the RKGC formulation. Fig. 27 illustrates the configurations of the wave tank and the OWSC model, which are identical to the experimental setup detailed in Ref. [84]. The wave tank measures 18.42 m in length, 4.58 m in width, and 1.0 m in height. The OWSC device is simplified as a flap with dimensions of 0.48 m in height, 1.04 m in width, and 0.12 m in thickness, and is hinged to a base with a height of 0.16 m. The flap has a mass of 33 kg, and its angular inertia is 1.84 kg/m<sup>2</sup>. To measure the wave elevation and impact pressure on the flap, three wave gauges, as shown in Fig. 27, and six pressure sensors, whose positions are listed in Table 2, are employed. Note that the present model is 1:25 scaled to that in the Ref. [84], and all the results present in this work have been converted to the full scale accordingly.

Considering the regular wave with a height  $H = 5$  m and a period  $T = 10$  s at full scale, a piston-type wave maker is employed to generate regular waves, adopting an ensemble of dummy particles whose motion follows the linear wavemaker theory [72],

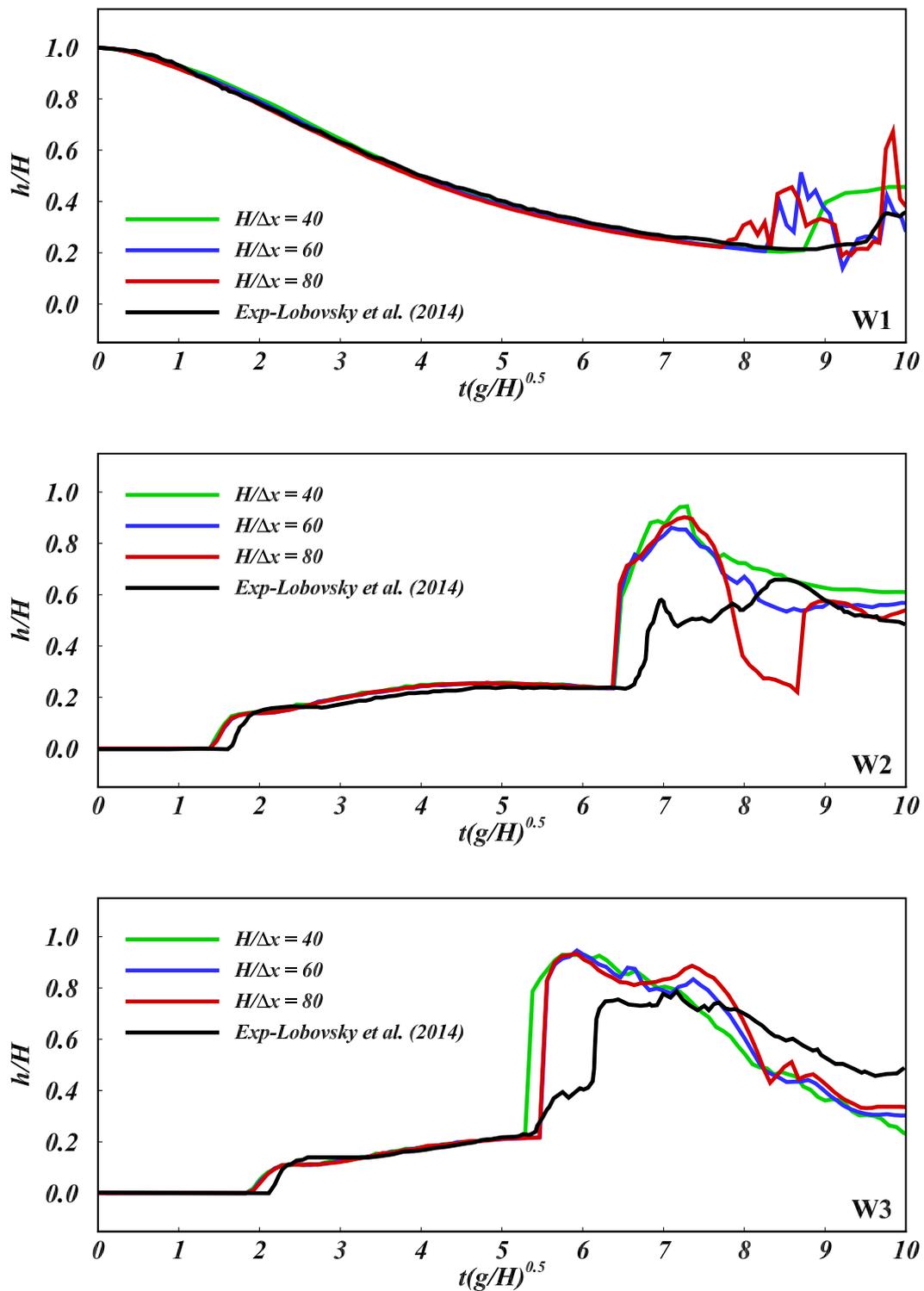


Fig. 24. Three-dimensional dam-break: water levels recorded at W1, W2, and W3. Convergence study and comparison against experimental results presented by Lobovsky et al. [76].

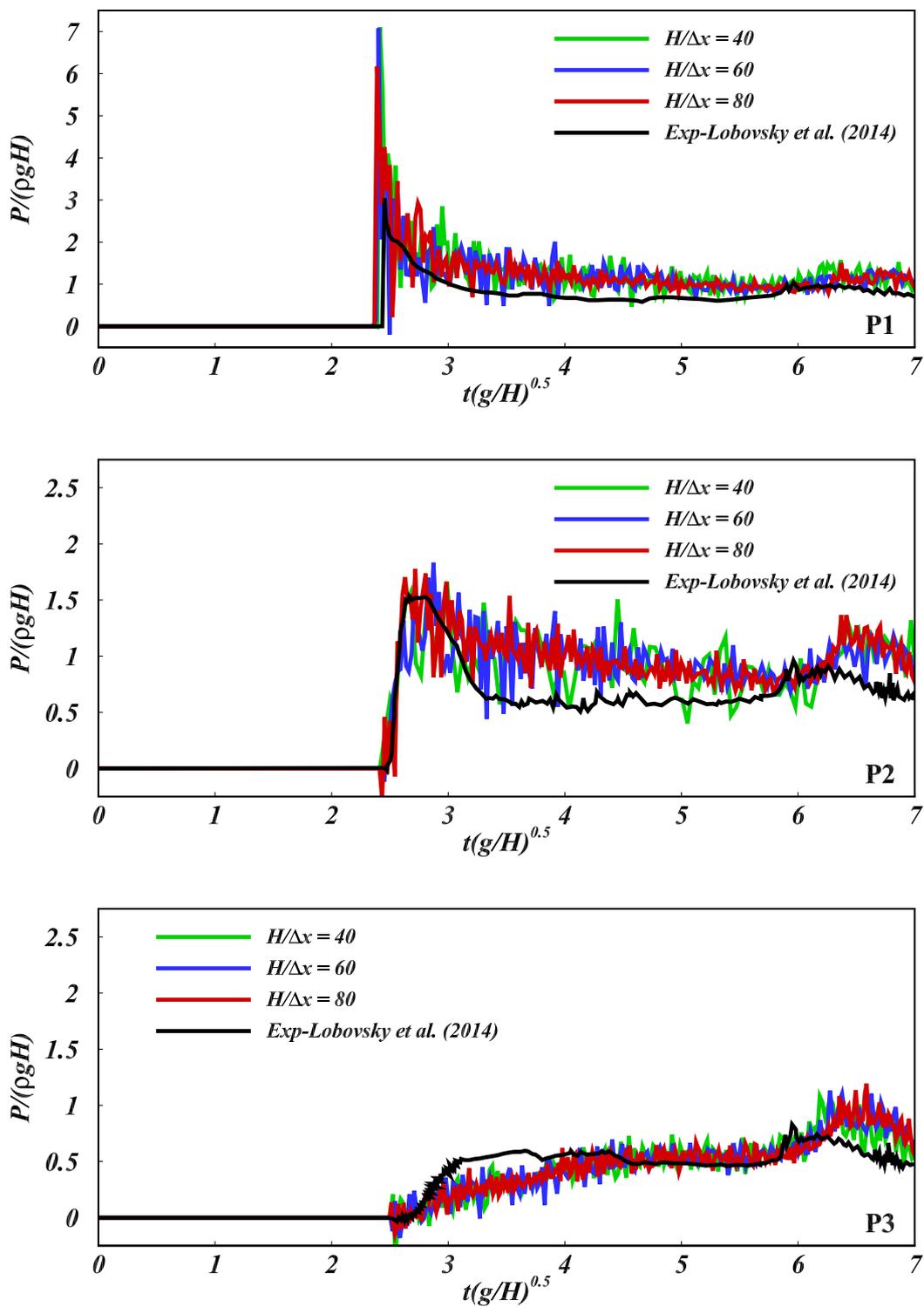
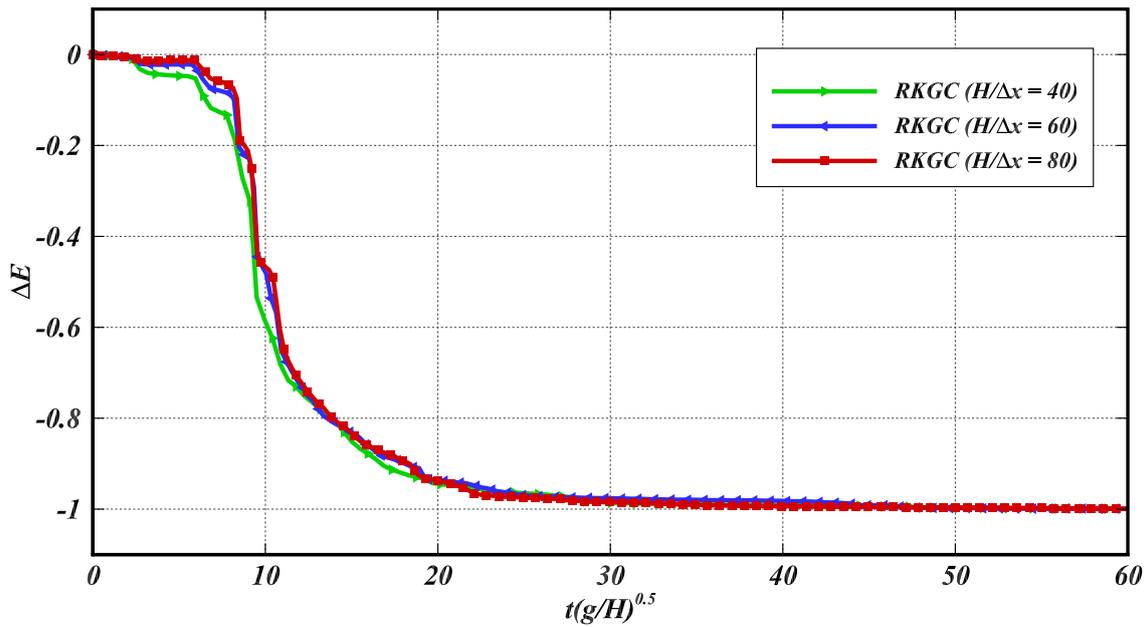
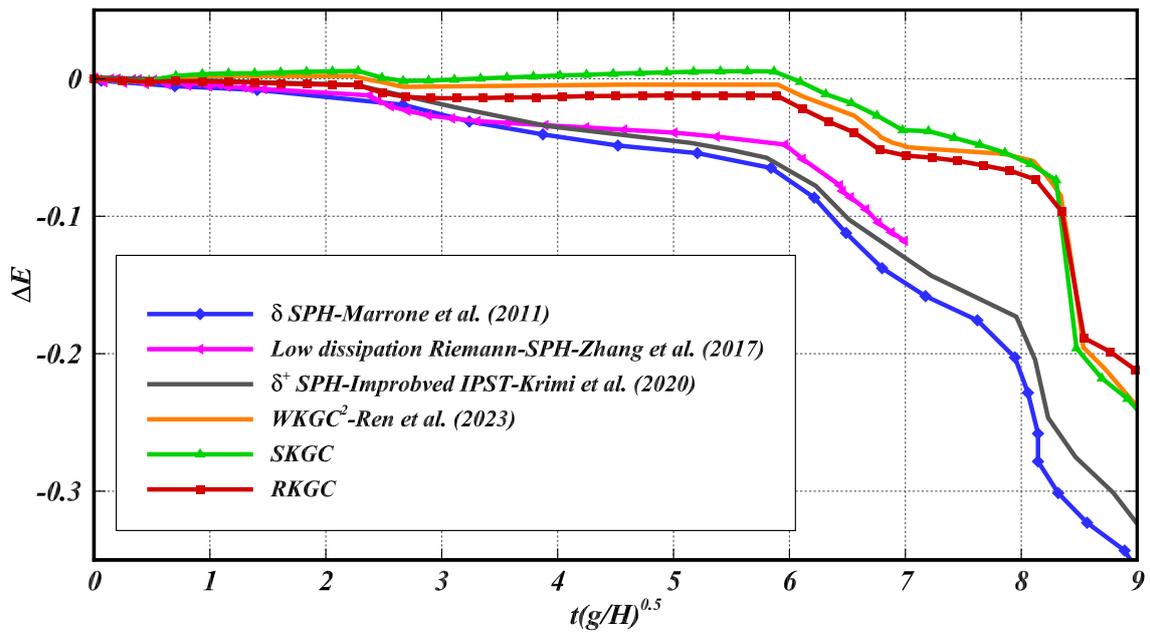


Fig. 25. Three-dimensional dam-break: history of pressure signals recorded at probes P1, P2 and P3. Convergence study and comparison against experimental results presented by Lobovsky et al. [76].



(a)



(b)

Fig. 26. Dam-break flow: evolution of mechanical energy. (a) Global evolution; (b) Zoom in on the initial evolution and compare with references ( $H/\Delta x = 80$ ).

where the particle displacement in  $x$ -direction  $x_a$  is determined by

$$x_a = S \sin(ft + \phi), \tag{44}$$

with  $S$  the wave stroke,  $f$  the wave frequency, and  $\phi$  the initial phase. The wave stroke is further defined as

$$S = \frac{H \sinh(2kh_0) + 2kh_0}{\sinh(2kh_0) \tanh(kh_0)}, \tag{45}$$

where  $h_0$  is the water depth, and  $k$  is the wave number. To minimize wave reflective effects on the pressure, wave energy, flap movements, etc., a damping zone [87,88] as illustrated in Fig. 27, is established at the end of the wave tank to absorb energy from

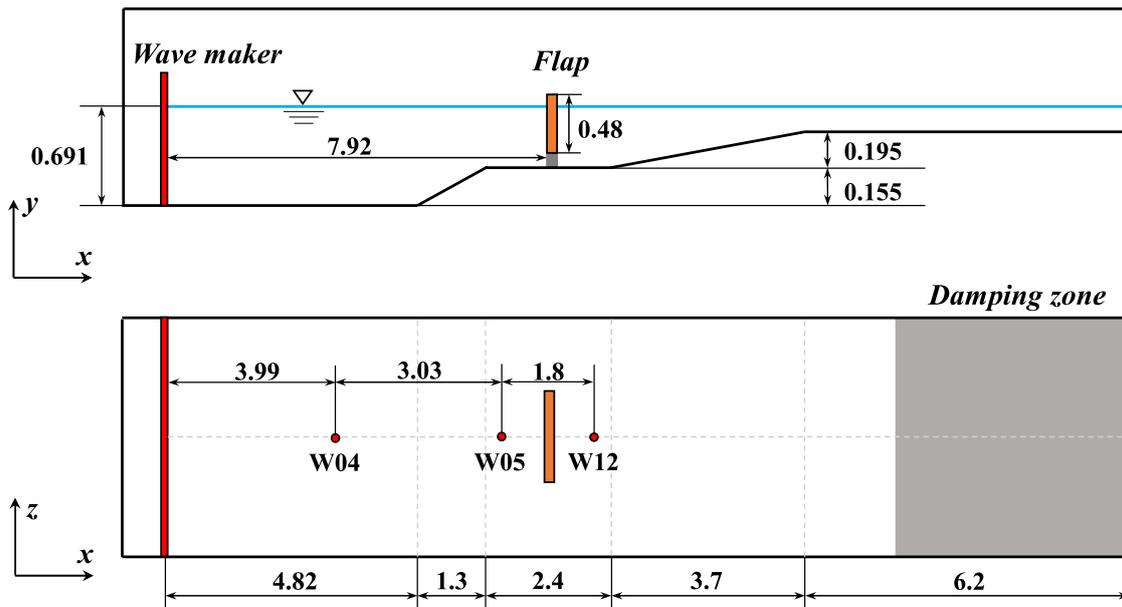


Fig. 27. OWSC: Schematic depiction of the wave tank and the OWSC model.

waves reaching its boundary and reduce the amplitude of reflected waves. Within this damping zone, the particle velocity undergoes decay according to

$$v = v_0 \left( 1.0 - \Delta t \theta \left( \frac{r - r_0}{r_1 - r_0} \right) \right) \tag{46}$$

Here,  $v_0$  the initial velocity of the fluid particle at the entry of the damping zone,  $v$  the velocity after damping,  $\Delta t$  the time step, and  $r_0$  and  $r_1$  are the initial and final positions of the damping zone, respectively. The reduction coefficient  $\theta = 5.0$  governs the modifications on the velocity at each time step in the current simulation. The entire system is discretized with a particle spacing of 0.03 m, resulting in 1.542 million fluid particles and 0.628 million solid particles. The present numerical results have been compared with both experimental observations and numerical investigations reported in Ref. [84].

Fig. 28 displays snapshots of the free-surface profile colored by the normalized pressure 28(a) and velocity magnitude 28(b), respectively. The results clearly demonstrate that the current method effectively captures the dynamic free-surface elevation, including wave reflection and breaking around the flap. Additionally, the outcomes exhibit smooth pressure and velocity fields, even during the intensive wave interactions around the flap, where wave reflection and breaking are observed. Furthermore, the cross-sectional slices along the middle line provide insights into the rotational state of the flap. These observations are consistent with those reported in the Refs. [84,86], as evidenced by the wave height and flap rotation angle history presented below.

The observed wave evaluations are depicted in Fig. 29, offering a comparison with the reference results [45,84]. The RKG formulation demonstrates good agreement with the results obtained from experiments, particularly at locations W04 and W05, which are in the seaward direction from the flap. However, discrepancies, especially the overestimation of wave crest height, are noticeable at location W12, positioned behind the flap. These differences may arise from wave reflection and breaking around the flap. Additionally, though the reference numerical results [84] utilize a turbulence model to introduce additional numerical dissipation, it is reasonable not to introduce the turbulence model, as investigated by He et al. [89]. Another possible reason for the discrepancies may be attributed to the single-phase flow simulation in the current study. The results from multi-phase flow, where the air phase is also considered by He et al. [89,90] can be closer to the actual case, and they conclude the gas plays a vital role in attenuating wave height during wave propagation. Compared to the results in Ref. [45] where WKGC<sup>2</sup> is adopted, RKG shows better alignment with experimental, particularly predicting more accurate wave heights at W04 and W05 positions and wave falls at all three positions.

Fig. 30 presents the rotation angle of the flap, providing a comparison with the experimental observations and other numerical predictions. The comparison underscores the good agreement with the experimental and Fluent results [84]. While predictions obtained by standard SPH formulations without corrections, such as those reported in Ref. [83,85], often underestimate the extreme rotation angle of the flap compared to experimental results, the prediction gained with the RKG formulation and WKGC<sup>2</sup> [45] notably overcomes this limitation. They provide more accurate predictions for flap rotating angles, aligning more closely with the experimental observations. The RKG formulation in this study is applied only in the fluid domain rather than the fluid–structure interaction, thus, it gives slightly closer predictions to experiments than that in Ref. [45].

The pressure evolution over time at each probe on the flap is illustrated and compared in Fig. 31. The results obtained using the RKG formulation align well with the experimental data, predicting reasonable slamming pressures despite some pressure oscillations due to the weakly compressible assumption. In addition, the observed large pressure peaks and drops at P01, P03, P09,

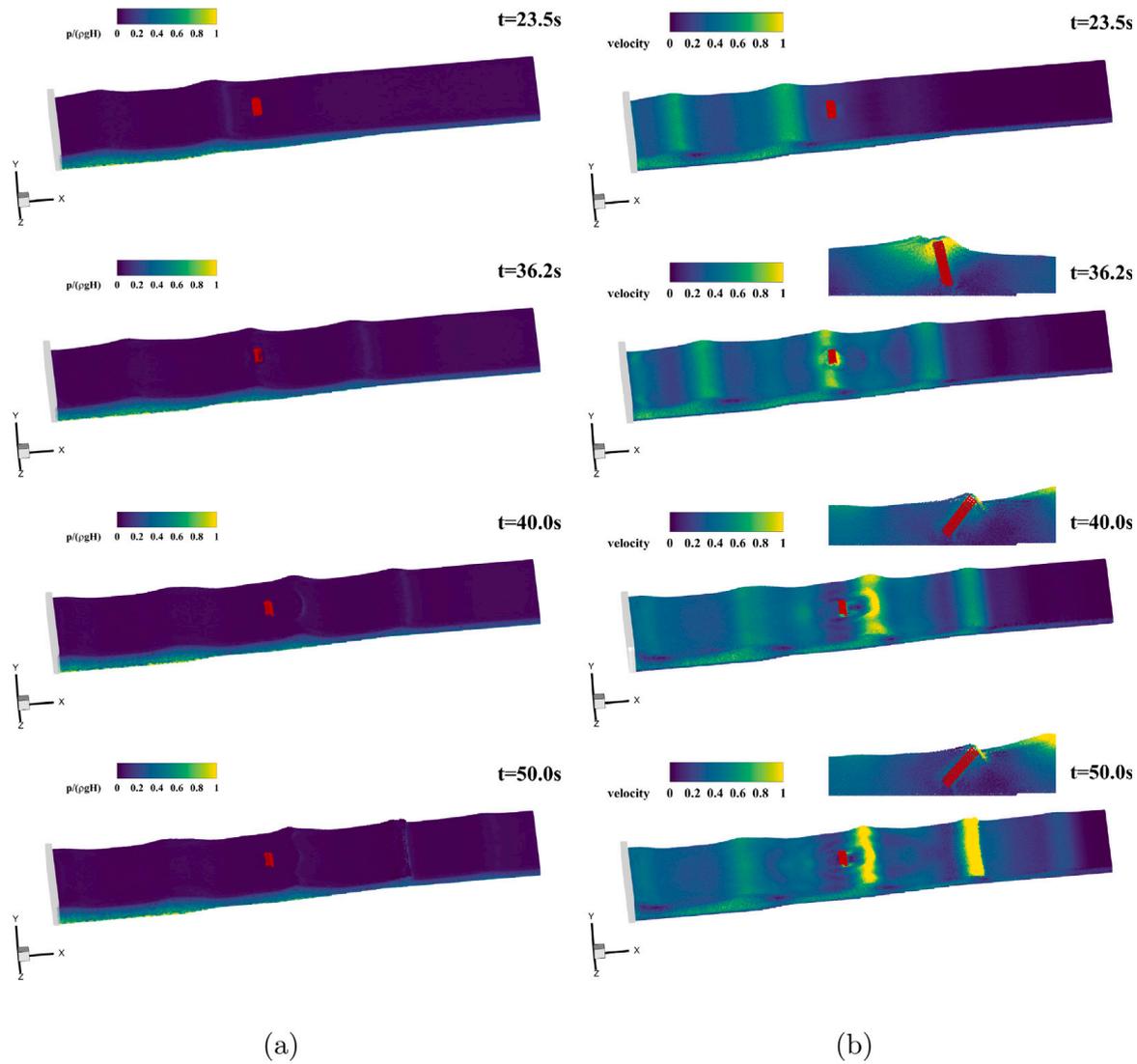


Fig. 28. OWSC: Snapshots of free surfaces and the flap motion during time evolution. (a) Fluid particles are colored by the normalized pressure; (b) Fluid particles are colored by the velocity magnitude. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and P11 can be attributed to the fact that air cushion effects are not captured in the current single-fluid simulation. Compared to the pressure results in Ref. [45,86] based on the SPH method, the current observed pressure shows fluctuations due to the density reinitialization method employed in Ref. [91], which has been proven more suitable for free-surface problems. The capturing of double pressure peaks at P01, P03, and P15, as well as the pressure drops at P05 and P13, show better agreement than those obtained using Fluent, although the small fluctuations are observed due to the weakly compressible model. However, discrepancies still exist, which may be attributed to air entrainment in the splash passing the flap, a factor not considered in the current work. Also, note that the pressure sensor P11 gives an apparent small non-zero reading before the wavefront arrives. This is because the sensor is quite close to the initial water level (0.025 m higher) within a particle spacing  $\Delta x = 0.03$  m and the cutoff radius for data sampling. The initial particle rearrangement at the early stage of the simulation can produce such small magnitude but erroneous signals. Later, these signals are replaced with more significant and physically meaningful values. This observation has also been reported in previous work [45].

In terms of computational cost for RKGC formulation, a comprehensive comparison of SKGC detailed in previous work [45] can be referred to. With the adoption of dual-time stepping techniques, there are around 15% additional computational costs for the 3D problem to calculate the KGC matrix and implement the corrected formulation.

## 6. Extension

The concept of the reverse KGC formulation can be readily extended to accommodate second- or even higher-order consistency conditions of the SPH gradient approximation. This extension uses the correction function for RKPM proposed by Liu et al. [21,22],

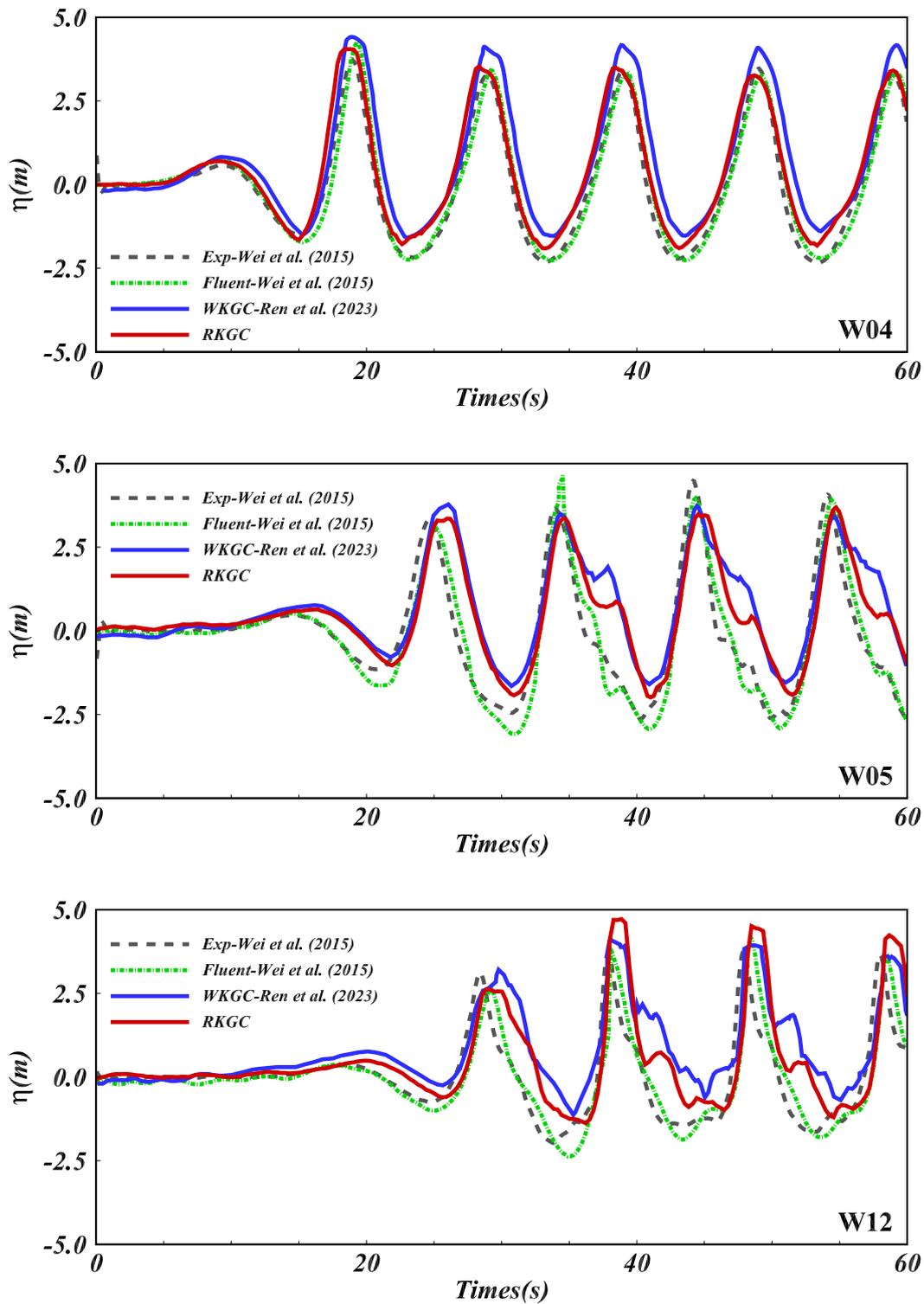


Fig. 29. OWSC: Comparison of free surface elevations for wave height  $H = 5.0$  m and period  $T = 10.0$  s against results of Wei et al. [84] and Ren et al. [45].

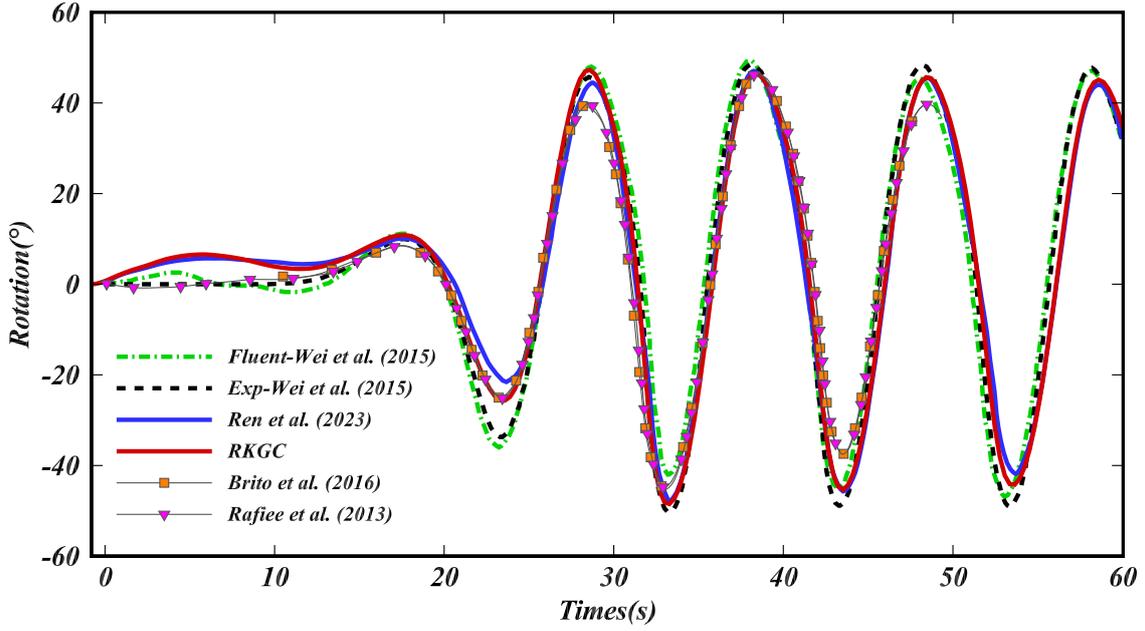


Fig. 30. OWSC: Comparison of the time evolution of the flap rotation.

provided that the corresponding particle relaxation is employed. The SPH gradient approximation in non-conservative form can be expressed similarly to Eq. (9) as

$$\nabla\psi_i = \sum_j \psi_{ij} \mathbf{C}_i(\mathbf{r}_i, \mathbf{r}_j) \nabla W_{ij} V_j, \quad (47)$$

where  $\mathbf{C}_i(\mathbf{r}_i, \mathbf{r}_j) = \mathbf{C}_0(\mathbf{r}_i) + \mathbf{C}_1(\mathbf{r}_i)(\mathbf{r}_j - \mathbf{r}_i)$  represents the correction function (see Refs. [21,22] for more definitions). Following Taylor expansion of  $\psi_j$  and substituting it into (47), one can obtain

$$\nabla\psi_i = - \sum_j \left( \nabla\psi_i \cdot \mathbf{r}_{ij} + \frac{1}{2} \nabla \cdot \nabla\psi_i : \mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \right) \mathbf{C}_i(\mathbf{r}_i, \mathbf{r}_j) \nabla W_{ij} V_j. \quad (48)$$

For vanishing leading moments and for ensuring second-order consistency, the following conditions are to be satisfied simultaneously as

$$\begin{cases} \sum_j -\mathbf{r}_{ij} \otimes \mathbf{C}_i(\mathbf{r}_i, \mathbf{r}_j) \nabla W_{ij} V_j = \mathbf{I} \\ \sum_j \mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \cdot \mathbf{C}_i(\mathbf{r}_i, \mathbf{r}_j) \nabla W_{ij} V_j = \mathbf{0}. \end{cases} \quad (49)$$

Subsequently,  $\mathbf{C}_0$  and  $\mathbf{C}_1$  can be obtained by solving Eq. (49). Consequently, the reverse-corrected conservative formulation with second-order consistency can be written similarly to Eq. (24) as

$$\nabla\psi_i = - \sum_j (\psi_i \mathbf{C}_j + \psi_j \mathbf{C}_i) \nabla W_{ij} V_j, \quad (50)$$

which can be further rewritten as

$$\nabla\psi_i = -\psi_i \sum_j (\mathbf{C}_i + \mathbf{C}_j) \nabla W_{ij} V_j + \sum_j \psi_{ij} \mathbf{C}_i \nabla W_{ij} V_j. \quad (51)$$

The first term on the RHS can also be made to vanish by particle relaxation based on the correction function, while the second term accurately reproduces the gradient as indicated in Eq. (47). Therefore, the reverse-corrected conservative formulation in Eq. (50) exhibits second-order consistency, and it also satisfies the discrete variational consistency and the linear momentum conservation, as we have discussed in Section 3.3. However, obtaining the corresponding correction function and achieving convergence in the particle relaxation driven by the correction function are still open issues.

## 7. Conclusion

This paper introduces the reverse KGC (RKGC) formulation, which is conservative and, integrating the particle relaxation based on the KGC matrix, ensures the zero- and first-order consistencies without explicit dependence on the smoothing length. Implementation in typical SPH methods, including Lagrangian SPH and Eulerian SPH, exhibits considerably improved accuracy, especially good energy conservation properties in general free-surface problems. While resulting the straightforward extension of the scheme to even

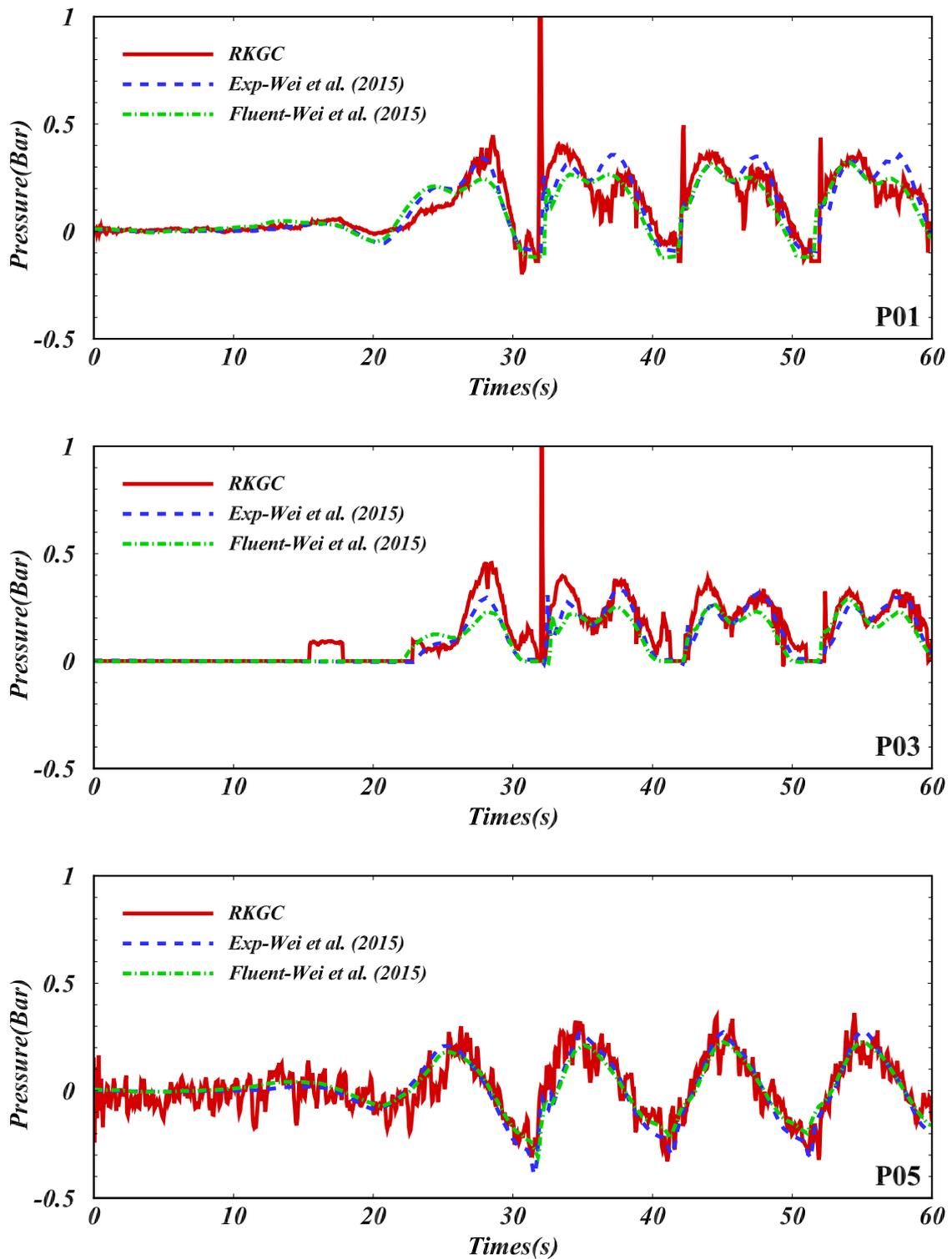


Fig. 31. OWSC: Comparison of pressure on the flap for  $H = 5.0$  m and  $T = 10.0$  s against the results of Wei et al. [84].

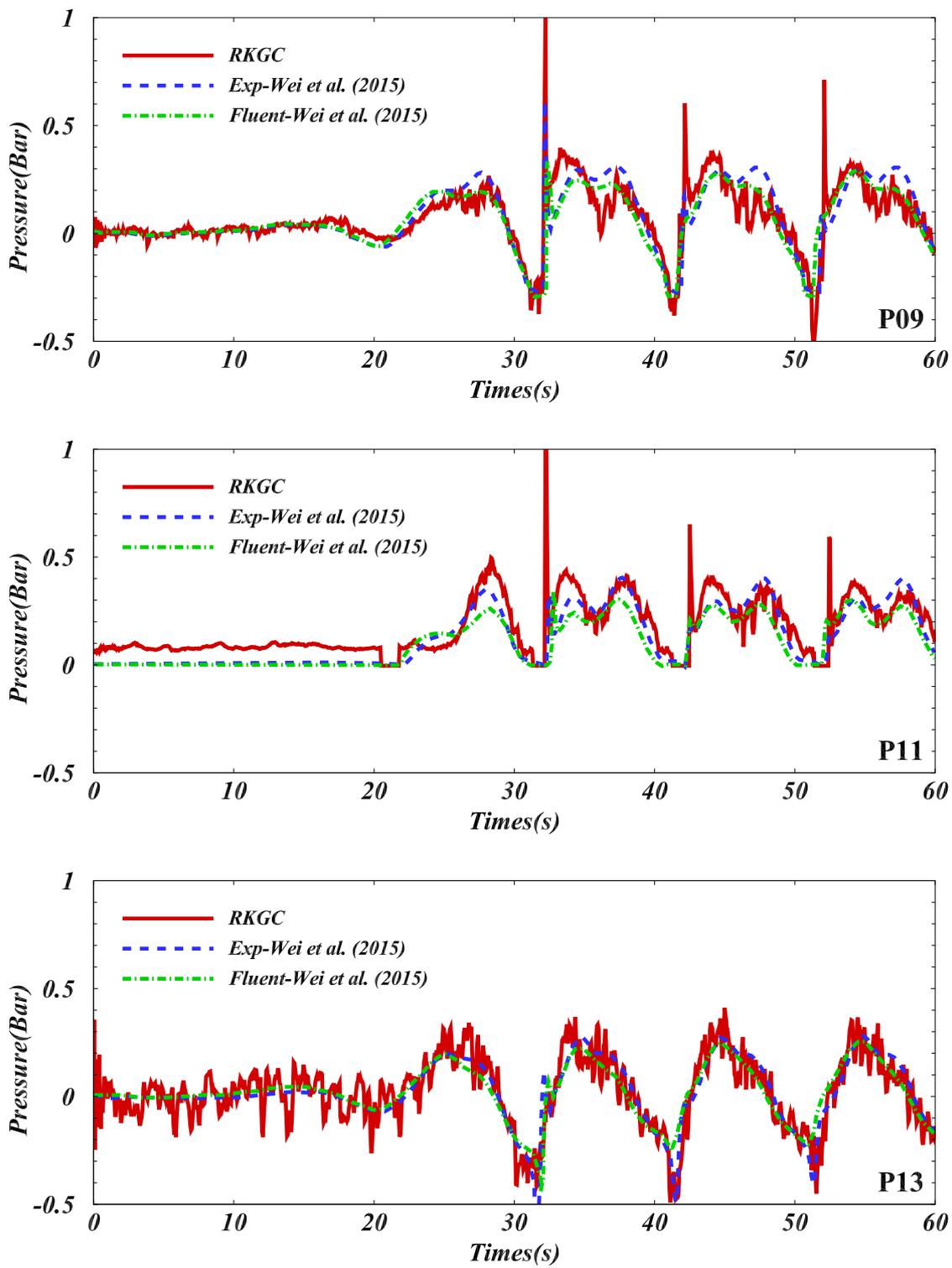


Fig. 31. (continued).

higher order consistency, challenges in achieving converged solutions for particle relaxation driven by the KGC matrix and high-order correction function are yet to be elaborated, especially for three-dimensional complex geometries and the situation employing a  $h/\Delta x$  value smaller than 1.0. Additionally, extending RKGC for SPH solid dynamics and to a similar idea for Laplacian operators is subjected to future work.

### CRediT authorship contribution statement

**Bo Zhang:** Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Nikolaus Adams:** Writing – review & editing, Investigation. **Xiangyu Hu:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, 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.

### Acknowledgments

Bo Zhang acknowledges the financial support provided by the China Scholarship Council (No. 202006230071). X.Y. Hu expresses gratitude to the Deutsche Forschungsgemeinschaft (DFG) for sponsoring this research under grant number DFG HU1527/12-4. The corresponding code for this work is available on GitHub at <https://github.com/Xiangyu-Hu/SPHinXsys>.

### Data availability

The code is open source.

### References

- [1] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (1977) 1013–1024, vol. 82, Dec. 1977, pp. 1013–1024.
- [2] 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.
- [3] J.J. Monaghan, Simulating free surface flows with SPH, *J. Comput. Phys.* 110 (2) (1994) 399–406.
- [4] X.Y. Hu, N.A. Adams, A multi-phase SPH method for macroscopic and mesoscopic flows, *J. Comput. Phys.* 213 (2) (2006) 844–861.
- [5] 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.
- [6] W. Benz, E. Asphaug, Simulations of brittle solids using smooth particle hydrodynamics, *Comput. Phys. Commun.* 87 (1–2) (1995) 253–265.
- [7] T. Rabczuk, J. Eibl, Simulation of high velocity concrete fragmentation using SPH/MLSPH, *Internat. J. Numer. Methods Engrg.* 56 (10) (2003) 1421–1444.
- [8] Y.-X. Peng, A.-M. Zhang, F.-R. Ming, S.-P. Wang, A meshfree framework for the numerical simulation of elasto-plasticity deformation of ship structure, *Ocean Eng.* 192 (2019) 106507.
- [9] T. Ye, D. Pan, C. Huang, M. Liu, Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications, *Phys. Fluids* 31 (1) (2019) 011301.
- [10] H. Gotoh, A. Khayyer, On the state-of-the-art of particle methods for coastal and ocean engineering, *Coast. Eng. J.* 60 (1) (2018) 79–103.
- [11] C. Zhang, M. Rezavand, X. Hu, A multi-resolution SPH method for fluid-structure interactions, *J. Comput. Phys.* 429 (2021) 110028.
- [12] D.A. Fulk, A Numerical Analysis of Smoothed Particle Hydrodynamics, Air Force Institute of Technology, 1994.
- [13] D.A. Fulk, D.W. Quinn, An analysis of 1-D smoothed particle hydrodynamics kernels, *J. Comput. Phys.* 126 (1) (1996) 165–180.
- [14] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *Internat. J. Numer. Methods Engrg.* 66 (13) (2006) 2064–2085.
- [15] S. Litvinov, X. Hu, N.A. Adams, Towards consistence and convergence of conservative SPH approximations, *J. Comput. Phys.* 301 (2015) 394–401.
- [16] Smoothed particle hydrodynamics, in: *Annual Review of Astronomy and Astrophysics*, Vol. 30, 30, 1992, pp. 543–574, (A93-25826 09-90), pp. 543–574.
- [17] V.G. Maz'ia, G. Schmidt, Approximate Approximations, no. 141, American Mathematical Soc., 2007.
- [18] G. Oger, M. Doring, B. Alessandrini, P. Ferrant, An improved SPH method: Towards higher order convergence, *J. Comput. Phys.* 225 (2) (2007) 1472–1492.
- [19] P. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, *Comput. Methods Appl. Mech. Engrg.* 139 (1–4) (1996) 375–408.
- [20] J. Chen, J. Beraun, T. Carney, A corrective smoothed particle method for boundary value problems in heat conduction, *Internat. J. Numer. Methods Engrg.* 46 (2) (1999) 231–252.
- [21] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Methods Fluids* 20 (8–9) (1995) 1081–1106.
- [22] W.K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics, *Internat. J. Numer. Methods Engrg.* 38 (10) (1995) 1655–1679.
- [23] S.N. Atluri, J. Cho, H.-G. Kim, Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations, *Comput. Mech.* 24 (5) (1999) 334–347.
- [24] M. Liu, W. Xie, G. Liu, Modeling incompressible flows using a finite particle method, *Appl. Math. Model.* 29 (12) (2005) 1252–1270.
- [25] M. Liu, G.-R. Liu, Restoring particle consistency in smoothed particle hydrodynamics, *Appl. Numer. Math.* 56 (1) (2006) 19–36.
- [26] Z. Zhang, M. Liu, A decoupled finite particle method for modeling incompressible flows with free surfaces, *Appl. Math. Model.* 60 (2018) 606–633.
- [27] Z. Zhang, K. Walayat, J. Chang, M. Liu, Meshfree modeling of a fluid-particle two-phase flow with an improved SPH method, *Internat. J. Numer. Methods Engrg.* 116 (8) (2018) 530–569.
- [28] C. Huang, D. Zhang, Y. Si, Y. Shi, Y. Lin, Coupled finite particle method for simulations of wave and structure interaction, *Coast. Eng. J.* 140 (2018) 147–160.
- [29] C. Huang, D. Zhang, Y. Shi, Y. Si, B. Huang, Coupled finite particle method with a modified particle shifting technology, *Internat. J. Numer. Methods Engrg.* 113 (2) (2018) 179–207.
- [30] R. Batra, G. Zhang, Analysis of adiabatic shear bands in elasto-thermo-viscoplastic materials by modified smoothed-particle hydrodynamics (MSPH) method, *J. Comput. Phys.* 201 (1) (2004) 172–190.

- [31] G. Zhang, R. Batra, Modified smoothed particle hydrodynamics method and its application to transient problems, *Comput. Mech.* 34 (2) (2004) 137–146.
- [32] S. Sibilla, An algorithm to improve consistency in smoothed particle hydrodynamics, *Comput. & Fluids* 118 (2015) 148–158.
- [33] 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.
- [34] C. Huang, J. Lei, M. Liu, X. Peng, A kernel gradient free (KGF) SPH method, *Internat. J. Numer. Methods Fluids* 78 (11) (2015) 691–707.
- [35] 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.
- [36] N. Flyer, B. Fornberg, Radial basis functions: Developments and applications to planetary scale flows, *Comput. & Fluids* 46 (1) (2011) 23–32.
- [37] J. King, S.J. Lind, A.M. Nasar, High order difference schemes using the local anisotropic basis function method, *J. Comput. Phys.* 415 (2020) 109549.
- [38] N. Trask, M. Perego, P. Bochev, A high-order staggered meshless method for elliptic problems, *SIAM J. Sci. Comput.* 39 (2) (2017) A479–A502.
- [39] 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.
- [40] S. Adami, X. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, *J. Comput. Phys.* 241 (2013) 292–307.
- [41] C. Zhang, X.Y. Hu, N.A. Adams, A generalized transport-velocity formulation for smoothed particle hydrodynamics, *J. Comput. Phys.* 337 (2017) 216–232.
- [42] 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.
- [43] J. Bonet, T.-S. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamic formulations, *Comput. Methods Appl. Mech. Engrg.* 180 (1–2) (1999) 97–115.
- [44] P.R.R. de Campos, A.J. Gil, C.H. Lee, M. Giacomini, J. Bonet, A new updated reference Lagrangian smooth particle hydrodynamics algorithm for isothermal elasticity and elasto-plasticity, *Comput. Methods Appl. Mech. Engrg.* 392 (2022) 114680.
- [45] 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. Engrg.* 417 (2023) 116460.
- [46] J.P. Vila, SPH renormalized hybrid methods for conservation laws: applications to free surface flows, in: *Meshfree Methods for Partial Differential Equations II*, Springer, 2005, pp. 207–229.
- [47] 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.
- [48] G. Liang, X. Yang, Z. Zhang, G. Zhang, Study on the propagation of regular water waves in a numerical wave flume with the  $\delta$ -SPHC model, *Appl. Ocean Res.* 135 (2023) 103559.
- [49] X. Huang, P. Sun, H. Lü, S. Zhong, Development of a numerical wave tank with a corrected smoothed particle hydrodynamics scheme to reduce nonphysical energy dissipation, *Chin. J. Theor. Appl. Mech.* 54 (6) (2022) 1502–1515.
- [50] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, *J. Comput. Phys.* 191 (2) (2003) 448–475.
- [51] C. Zhang, M. Rezavand, X. Hu, Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics, *J. Comput. Phys.* 404 (2020) 109135.
- [52] J.P. Vila, On particle weighted methods and smooth particle hydrodynamics, *Math. Models Methods Appl. Sci.* 9 (02) (1999) 161–209.
- [53] Z. Wang, C. Zhang, O.J. Haidn, N.A. Adams, X. Hu, Extended Eulerian SPH and its realization of FVM, 2023, arXiv preprint arXiv:2309.01596.
- [54] 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.
- [55] F. Rieper, On the dissipation mechanism of upwind-schemes in the low Mach number regime: A comparison between Roe and HLL, *J. Comput. Phys.* 229 (2) (2010) 221–232.
- [56] H.-G. Lyu, P.-N. Sun, P.-Z. Liu, X.-T. Huang, A. Colagrossi, Derivation of an improved smoothed particle hydrodynamics model for establishing a three-dimensional numerical wave tank overcoming excessive numerical dissipation, *Phys. Fluids* 35 (6) (2023).
- [57] J.J. Monaghan, Smoothed particle hydrodynamics, *Rep. Prog. Phys.* 68 (8) (2005) 1703.
- [58] D.J. Price, Modelling discontinuities and Kelvin–Helmholtz instabilities in SPH, *J. Comput. Phys.* 227 (24) (2008) 10040–10057.
- [59] A. Colagrossi, M. Antuono, D. Le Touzé, Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model, *Phys. Rev. E* 79 (5) (2009) 056701.
- [60] P.W. Cleary, Modelling confined multi-material heat and mass flows using SPH, *Appl. Math. Model.* 22 (12) (1998) 981–993.
- [61] X. Hu, N. Adams, Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows, *Phys. Fluids* 18 (10) (2006).
- [62] P. Koumoutsakos, Multiscale flow simulations using particles, *Annu. Rev. Fluid Mech.* 37 (2005) 457–487.
- [63] 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.
- [64] Y. Zhu, C. Zhang, X. Hu, A consistency-driven particle-advection formulation for weakly-compressible smoothed particle hydrodynamics, *Comput. & Fluids* 230 (2021) 105140.
- [65] M. Bayareh, A. Nourbakhsh, F. Rouzbahani, V. Jouzaei, Explicit incompressible SPH algorithm for modelling channel and lid-driven flows, *SN Appl. Sci.* 1 (2019) 1–13.
- [66] G. Wu, R.E. Taylor, Finite element analysis of two-dimensional non-linear transient water waves, *Appl. Ocean Res.* 16 (6) (1994) 363–372.
- [67] A. Khayyer, H. Gotoh, Y. Shimizu, K. Gotoh, On enhancement of energy conservation properties of projection-based particle methods, *Eur. J. Mech. B Fluids* 66 (2017) 20–37.
- [68] A. Khayyer, Y. Shimizu, T. Gotoh, H. Gotoh, Enhanced resolution of the continuity equation in explicit weakly compressible SPH simulations of incompressible free-surface fluid flows, *Appl. Math. Model.* 116 (2023) 84–121.
- [69] M. Wang, Y. Pan, X. Shi, J. Wu, P. Sun, Comparative study on volume conservation among various SPH models for flows of different levels of violence, *Coast. Eng.* 191 (2024) 104521.
- [70] M. Antuono, S. Marrone, A. Colagrossi, B. Bouscasse, Energy balance in the  $\delta$ -SPH scheme, *Comput. Methods Appl. Mech. Engrg.* 289 (2015) 209–226.
- [71] J.J. Monaghan, A. Rafiee, A simple SPH algorithm for multi-fluid flow with high density ratios, *Internat. J. Numer. Methods Fluids* 71 (5) (2013) 537–561.
- [72] R.G. Dean, R.A. Dalrymple, *Water Wave Mechanics for Engineers and Scientists*, vol. 2, World Scientific Publishing Company, 1991.
- [73] J. Martin, W. Moyce, An experimental study of the collapse of fluid columns on a rigid horizontal plane, in a medium of lower, but comparable, density, *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 244 (882) (1952) 325–334.
- [74] B. Buchner, *Green Water on Ship-Type Offshore Structures* (Ph.D. thesis), Delft University of Technology Delft, The Netherlands, 2002.
- [75] T.-h. Lee, Z. Zhou, Y. Cao, Numerical simulations of hydraulic jumps in water sloshing and water impacting, *J. Fluids Eng.* 124 (1) (2002) 215–226.
- [76] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, *J. Fluids Struct.* 48 (2014) 407–434.
- [77] A. Ferrari, M. Dumbser, E.F. Toro, A. Armanini, A new 3D parallel SPH scheme for free surface flows, *Comput. & Fluids* 38 (6) (2009) 1203–1217.
- [78] J.L. Cercos-Pita, AQUAgusph, a new free 3D SPH solver accelerated with OpenCL, *Comput. Phys. Comm.* 192 (2015) 295–312.
- [79] A. Ritter, Die fortpflanzung der wasserwellen, *Z. Vereines Deutscher Ingen.* 36 (33) (1892) 947–954.
- [80] 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.
- [81] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, G. Graziani,  $\delta$ -SPH model for simulating violent impact flows, *Comput. Methods Appl. Mech. Engrg.* 200 (13–16) (2011) 1526–1542.
- [82] A. Krimi, M. Jandaghian, A. Shakibaenia, A WSPH particle shifting strategy for simulating violent free surface flows, *Water* 12 (11) (2020) 3189.

- [83] A. Rafiee, B. Elsaesser, F. Dias, Numerical simulation of wave interaction with an oscillating wave surge converter, in: *International Conference on Offshore Mechanics and Arctic Engineering*, Vol. 55393, American Society of Mechanical Engineers, 2013, V005T06A013.
- [84] Y. Wei, A. Rafiee, A. Henry, F. Dias, Wave interaction with an oscillating wave surge converter, Part I: Viscous effects, *Ocean Eng.* 104 (2015) 185–203.
- [85] M. Brito, R. Canelas, R. Ferreira, O. García-Feal, J. Domínguez, A. Crespo, M. Neves, Coupling between DualSPHysics and Chrono-Engine: towards large scale HPC multiphysics simulations, in: *11th International SPHERIC Workshop*, Munich, Germany, 2016.
- [86] C. Zhang, Y. Wei, F. Dias, X. Hu, An efficient fully Lagrangian solver for modeling wave interaction with oscillating wave surge converter, *Ocean Eng.* 236 (2021) 109540.
- [87] S.J. Lind, R. Xu, P.K. Stansby, B.D. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, *J. Comput. Phys.* 231 (4) (2012) 1499–1523.
- [88] X. Weng, C. Huang, T. Long, S. Li, M. Liu, An integrated finite particle method with perfectly matched layer for modeling wave-structure interaction, *Coast. Eng. J.* 61 (1) (2019) 78–95.
- [89] F. He, Y. Zhang, H. Jiang, C. Huang, Numerical investigation of solitary wave breaking over a slope based on multi-phase smoothed particle hydrodynamics, *Phys. Fluids* 35 (2) (2023).
- [90] F. He, H. Jiang, Y. Lin, J. Pan, Y. Zhang, C. Huang, Multi-phase SPH-FDM and experimental investigations on the hydrodynamics of an oscillating water column wave energy device, *Coast. Eng.* (2024) 104569.
- [91] N. Salis, X. Hu, M. Luo, A. Reali, S. Manenti, 3D SPH analysis of focused waves interacting with a floating structure, *Appl. Ocean Res.* 144 (2024) 103885.



## A.2 Paper II

Bo Zhang, Chi Zhang, Xiangyu Hu

### **Target-driven splitting SPH optimization of thermal conductivity distribution**

In *International Journal of Heat and Mass Transfer*, Volume 227: 125476, 2024, DOI: <https://doi.org/10.1016/j.ijheatmasstransfer.2024.125476>.

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.





# Target-driven splitting SPH optimization of thermal conductivity distribution

Bo Zhang<sup>a</sup>, Chi Zhang<sup>b</sup>, Xiangyu Hu<sup>a,\*</sup>

<sup>a</sup> TUM School of Engineering and Design, Technical University of Munich, 85748 Garching, Germany

<sup>b</sup> Huawei Technologies Munich Research Center, 80992 Munich, Germany

## ARTICLE INFO

### Keywords:

Heat transfer enhancement  
Volume-to-point problem  
PDE-constrained optimization  
Target-driven approach  
Gradient descent method  
Smoothed particle hydrodynamics  
Implicit splitting operator

## ABSTRACT

Efficiently enhancing heat conduction through optimized distribution of a limited quantity of high thermal conductivity material is paramount in cooling electronic devices and numerous other applications. This paper introduces a target-driven all-at-once approach for PDE-constrained optimization and derives a splitting smoothed particle hydrodynamics (SPH) method for optimizing the distribution of thermal conductivity in heat conduction problems. In this method, the optimization iteration of the system is split into several easily addressed steps. A targeting step is employed to progressively enforce the direct target, which potentially leads to increased PDE residuals. Then, these residuals are recovered through an evolution step of the design variable. After this, a PDE solution step is carried out to further decrease the PDE residuals, and the system is ready for the next iteration. Unlike the simulation-based approaches, the present method does not rely on the adjoint state equation and converged state variable field in each iteration, and the optimization process is significantly simplified and accelerated. With the utilization of an implicit SPH splitting operator and a general numerical regularization formulation, the information propagation is further accelerated and the numerical stability is greatly enhanced. Typical examples of heat conduction optimization demonstrate that the current method yields optimal results comparable to previous methods and exhibits considerable computational efficiency. Moreover, the optimal results feature more moderate extreme values, which offers distinct advantages for the easier selection of appropriate material with high thermal conductivity.

## 1. Introduction

Electronic devices find wide-ranging applications in various industries, including aerospace, transportation, network communication, processing, and manufacturing. The evolution of microelectronics has facilitated the miniaturization of electronic devices, but this reduction in device size drastically increases power density and presents considerable challenges for effective cooling [1]. To guarantee the expected performance and lifespan, multiple design ideas have been explored to dissipate the generated heat and lower the operating temperatures [2–7]. Among these ideas, inserting high thermal conductivity material [2,3] has garnered extensive attention due to its effectiveness and simplicity of implementation. The rapid innovation in additive manufacturing, specifically in metal 3D printing, has also opened up new possibilities for implementing this cooling concept [8,9].

Thus, the primary design challenge lies in finding the optimal distributions of high thermal conductivity materials to meet specific cooling

targets. This optimization is also known as the volume-to-point (VP) problem [10]. It involves redistributing a fixed amount (constraint) of material with high thermal conductivity (design variable) to cool a heat-generating volume within specified boundaries, with the objective of minimizing the temperature (state variable) in the given domain. The VP problem is a typical example of PDE-constrained optimization, characterized by the constraint of physical principles expressed as partial differential equations (PDEs) [11].

The methods for PDE-constrained optimization can be categorized into simulation-based and all-at-once approaches, depending on how the PDE constraint is handled [12,13]. In simulation-based methods, it is eliminated by obtaining a converged physical solution using existing solvers. Therefore, the optimization process computes the gradients of state variables with respect to the design ones at the hypersurface of the physical solution to determine the searching direction in each iteration. Typical approaches for computing gradients, whether done explicitly or implicitly, include the adjoint technique [14,15], auto-

\* Corresponding author.

E-mail addresses: [bo.zhang.aer@tum.de](mailto:bo.zhang.aer@tum.de) (B. Zhang), [c.zhang@tum.de](mailto:c.zhang@tum.de) (C. Zhang), [xiangyu.hu@tum.de](mailto:xiangyu.hu@tum.de) (X. Hu).

<https://doi.org/10.1016/j.ijheatmasstransfer.2024.125476>

Received 4 December 2023; Received in revised form 26 February 2024; Accepted 20 March 2024

Available online 10 April 2024

0017-9310/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

matic differentiation (AD) [16] and artificial neural network (ANN) [17,18]. Although the simulation-based methods are conceptually appealing and widely used in solving VP problems [19–21], they require repeated and costly solutions of the PDEs, even in the initial stages when the design variables are still far from their optimal values. All-at-once methods, which explicitly maintain the PDE constraint as another optimal target and treat both the state and design variables equally. A clear advantage is that they do not require repeated PDE solutions but satisfy the PDE constraint only at the termination of optimization. Some all-at-once methods, such as the augmented Lagrangian method [22] and sequential quadratic programming method [23], have been proposed for addressing general PDE-constrained optimization problems. However, these methods are rare in solving VP problems due to their inherent complexity and high computational costs.

In early works on simulation-based methods for tackling the VP problem, some indirect global optimal principles, such as temperature gradient field homogenization [24–26], entropy generation minimization (EGM) [27–30], and entransy dissipation extremum (EDE) [26,31–34], have been proposed and adopted. Although these principles were expected to be equivalent to the direct target of the lowest average temperature, it has been argued that they are not equivalent and directly utilizing these principles for different targets or boundaries may lead to sub-optimal or even detrimental solutions [32]. Therefore, direct optimization targets, including the lowest average [35,15] and minimizing hot spot temperature [18,36], have also been employed recently.

In addition to the aforementioned iterative methods based on deterministic principles, some stochastic approaches, such as the bionic optimization (BO) method [37,38], cellular optimization (CA) method [39,40], simulated annealing (SA) and genetic algorithm (GA) method [41], have been explored to address VP problems. Topology optimization [42], for example the asymptotes algorithm (MMA) [43] and density-based method [44], has also gained attention for optimal heat conduction problems. While these stochastic methods offer novel avenues for achieving optimal solutions, they often deal with a large number of variables in the spatial discretization of the domain, leading to reduced efficiency and suitability for large-scale and reliability-sensitive problems. Therefore, developing an efficient yet straightforward method, particularly leveraging the direct optimization target and the all-at-once concept, holds great promise for addressing heat conduction optimization problems.

With these in mind, this paper introduces a target-driven all-at-once method for PDE-constrained optimization problems and derives a smoothed particle hydrodynamics (SPH) method for optimizing the thermal conductivity distribution to minimize the average temperature. In this method, the optimization iteration is divided into several easily handled steps. A targeting step is adopted to progressively impose the direct target, potentially resulting in increased PDE residuals. Then, through an evolution step of the design variables, these residuals are subsequently recovered. Following this, a PDE solution step is performed to further decrease the PDE residual and prepare for the next iteration. The novelty of this work can be summarized in three aspects. Firstly, as the split steps are only weakly coupled with each other, compared to previous all-at-once approaches, the present updating of both state and design variables is much easier to handle. Secondly, by leveraging the splitting-operator SPH method, implicit updating is achieved without the inversion of large-size matrices. Thirdly, a general formulation of regularization has been proposed to achieve numerical stability when evolving the design variable.

In the following sections, Section 2 provides a brief problem description; Section 3 introduces the SPH method and the numerical scheme; Section 4 elucidates the proposed optimization method, and Section 5 demonstrates typical examples to validate the effectiveness of the method. Finally, Section 6 concludes the key findings and outlooks the future work.

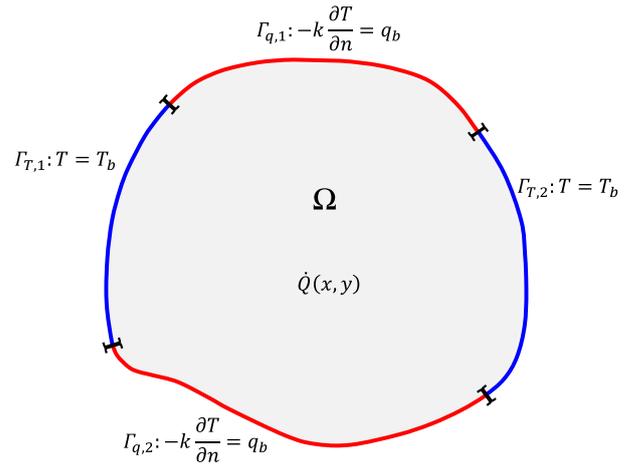


Fig. 1. Illustration of the 2D heat conduction problem.  $\Omega$  denotes the thermal domain;  $\Gamma_T$  and  $\Gamma_q$  represent constant surface temperature and heat flux boundaries, respectively, and  $\dot{Q}$  indicates the presence of an internal heat source.

## 2. Problem description

We consider the optimization of thermal conductivity distribution for two-dimensional heat conduction problems, specifically addressing typical VP problems. As illustrated in Fig. 1, the thermal domain under consideration, denoted as  $\Omega$ , is subject to different boundary conditions and may contain internal heat sources. The steady-state temperature field within the thermal domain can be determined by solving the transient heat conduction governing equation given by

$$\frac{dT}{dt} = \nabla \cdot (k \nabla T) + \dot{Q} \quad \text{in } \Omega, \quad (1)$$

when the left-hand side (LHS) converges to zero. Here,  $T$  represents temperature,  $k$  denotes thermal conductivity, and  $\dot{Q}$  is the volume rate of the internal heat source. Note that, for the sake of simplified analysis, Eq. (1) is reduced from the general heat conduction equation with  $\rho C = 1$ , where  $\rho$  and  $C$  represent density and heat capacity, respectively.

Two typical boundary conditions are considered here, as shown in Fig. 1. The Dirichlet boundary condition is given as

$$T = T_b \quad \text{on } \Gamma_T, \quad (2)$$

where the surface temperature  $T_b$  is held constant. The Neumann boundary condition, which maintains a fixed heat flux rate, can be expressed as,

$$-k \nabla T \cdot \mathbf{n} = q_b \quad \text{on } \Gamma_q, \quad (3)$$

where  $\mathbf{n}$  indicates the surface normal vector pointing outward.

The objective of the optimization is to obtain a distribution of thermal conductivity in  $\Omega$  by minimizing the average steady temperature  $\bar{T}$  defined as

$$\bar{T} = \frac{1}{V} \int_V T dV, \quad (4)$$

where  $V$  is the total volume of the thermal domain. Additionally, the average thermal conductivity is constrained to remain constant by

$$\int_V k dV = k_0 V, \quad (5)$$

where  $k_0$  is a reference value throughout the optimization process.

### 3. Numerical scheme

#### 3.1. SPH formulation

In this study, the SPH method is employed to solve the temperature and thermal conductivity fields. SPH is a fully Lagrangian particle method that was initially proposed for astrophysical applications [45,46]. Since its introduction, SPH has demonstrated significant success in simulating a wide range of scientific problems, including heat transfer problems [47,48].

In the SPH scheme, the heat conduction governing equation in Eq. (1) can be discretized at each SPH particle  $i$  located at  $\mathbf{r}_i$  with its neighboring particles  $j$  as follows,

$$\frac{dT_i}{dt} = 2 \sum_j \bar{k}_{ij} \frac{T_{ij}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i. \quad (6)$$

Here,  $\nabla_i W_{ij} = \nabla_i W(|\mathbf{r}_{ij}|, h) = \frac{\partial W_{ij}}{\partial r_{ij}} \mathbf{e}_{ij}$ , where  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ ,  $h$  is the smoothing length and the unit vector  $\mathbf{e}_{ij} = \frac{\mathbf{r}_{ij}}{r_{ij}}$ , represents the derivative of the kernel function.  $T_{ij} = T_i - T_j$  indicates the inter-particle temperature difference, and  $V_j$  denotes the volume of neighboring particles  $j$ . The expression  $\bar{k}_{ij} = (k_i + k_j)/2$  denotes the inter-particle average thermal conductivity in the context of a continuous thermal conductivity distribution.

Near the domain boundary, we introduce several layers of dummy particles (typically four layers, considering the currently selected smoothing length of 1.15) to enforce different boundary conditions. Implementing the Dirichlet boundary condition is straightforward and involves imposing the temperatures

$$T_w = 2T_b - T_i, \quad (7)$$

at dummy particles implied by the wall boundary condition [49]. To implement the Neumann boundary condition, the discretization of Eq. (1) is modified into

$$\frac{dT}{dt} = \nabla \cdot (k \nabla T) + \dot{Q} + \dot{Q}^{Fq} \quad \text{in } \Omega, \quad (8)$$

following the Ref. [50], where the heat flux in Eq. (3) is replaced by a volumetric term  $\dot{Q}^{Fq}$ , which can be discretized as

$$\dot{Q}_i^{Fq} = -q_b \sum_{j \in \Omega_j} (\mathbf{n}_i + \mathbf{n}_j) \cdot \nabla_i W_{ij} V_j. \quad (9)$$

Here,  $\Omega_j$  represents the boundary domain defined by the dummy particles. The unit vectors  $\mathbf{n}_i$  and  $\mathbf{n}_j$  are normal to the boundary evaluated at the respective positions of particle  $i$  and  $j$ .

#### 3.2. Splitting operator based implicit scheme

It is well known that traditional implicit schemes often require large-scale matrix inversion or iterations across the entire system, which can lead to significant memory demands and challenges in parallelization. To overcome these challenges, we employ a splitting operator based implicit scheme to advance Eq. (6). The implicit solving step is divided into individual particle-by-particle operations, and each evolves a small system that is easy to inverse. One commonly used approach for this purpose is the second-order Strang splitting technique [51], shown as

$$S_i^{(\Delta t)} = D_1^{(\frac{\Delta t}{2})} \circ D_2^{(\frac{\Delta t}{2})} \circ \dots \circ D_i^{(\frac{\Delta t}{2})} \circ \dots \circ D_{N_i-1}^{(\frac{\Delta t}{2})} \circ D_{N_i}^{(\frac{\Delta t}{2})} \circ D_{N_i}^{(\frac{\Delta t}{2})} \circ D_{N_i-1}^{(\frac{\Delta t}{2})} \circ \dots \circ D_i^{(\frac{\Delta t}{2})} \circ \dots \circ D_2^{(\frac{\Delta t}{2})} \circ D_1^{(\frac{\Delta t}{2})}. \quad (10)$$

Here, the operator  $S_i^{(\Delta t)}$  represents the complete step for advancing the equation.  $N_i$  refers to the total number of particles, and  $D_i$  represents the splitting operator corresponding to particle  $i$ . The update of the variable for the entire field involves a forward sweep of all particles for

half a time step, followed by a backward sweep for another half time step [52].

Within the local implicit formulation, Eq. (6) can be rewritten as

$$\frac{dT_i}{dt} = 2 \sum_j \bar{k}_{ij} \frac{T_{ij}^{n+1}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i^{n+1}, \quad (11)$$

where  $T_{ij}^{n+1} = T_{ij}^n + dT_i - dT_j$ . The terms  $dT_i$  and  $dT_j$  represent the incremental changes for particle  $i$  and its neighboring particles  $j$  at each advancing time step. For brevity, we introduce the coefficient

$$B_j = 2 \bar{k}_{ij} \frac{1}{r_{ij}} \nabla_i W_{ij} V_j dt, \quad (12)$$

and the residual of Eq. (11), without considering the increment, has the form

$$E_i = - \sum_j B_j T_{ij}^n - \dot{Q}_i^{n+1} dt. \quad (13)$$

The implicit formulation of Eq. (11) can be further expressed as

$$E_i = \left( \sum_j B_j - 1 \right) dT_i - \sum_j B_j dT_j. \quad (14)$$

To determine the incremental changes for temperature, we employ the gradient descent method [53] by reducing the LHS of Eq. (13) following its gradient. The gradient  $\nabla E_i$  with respect to the variable  $(dT_i, dT_1, dT_2, \dots, dT_N)^T$ , where  $N$  gives total number of all neighboring particles, can be obtained as

$$\nabla E_i = \left( \sum_j B_j - 1, -B_1, -B_2, \dots, -B_N \right)^T. \quad (15)$$

We set

$$(dT_i, dT_1, dT_2, \dots, dT_N)^T = \eta_i \nabla E_i, \quad (16)$$

where  $\eta_i$  represents the learning rate [53] for the particle  $i$ . Substituting Eqs. (15) and (16) into Eq. (14), the learning rate can be obtained as

$$\eta_i = \left( \left( \sum_j B_j - 1 \right)^2 + \sum_j (B_j)^2 \right)^{-1} E_i. \quad (17)$$

According to Eqs. (15) and (16), the incremental change in temperature of particle  $i$  and all its neighbors can be obtained and updated as

$$\begin{cases} T_i^{n+1} &= T_i^n - dT_i = T_i^n + \eta_i \left( \sum_j B_j - 1 \right) \\ T_1^{n+1} &= T_1^n - dT_1 = T_1^n - \eta_i B_1 \\ T_2^{n+1} &= T_2^n - dT_2 = T_2^n - \eta_i B_2 \\ &\dots \\ T_N^{n+1} &= T_N^n - dT_N = T_N^n - \eta_i B_N \end{cases}. \quad (18)$$

Note that, Eq. (18) involves updating the variables for particle  $i$  and its neighboring particles simultaneously. When a shared-memory parallelization is employed, conflict may arise when multiple threads attempt to update the values of a single particle pair simultaneously. To address this issue, we have implemented a splitting Cell Linked List method [54]. This method effectively prevents conflicts by ensuring that neighboring particles are located in the same cell or in adjacent cells that are distributed among the same threads. Also note that, for an explicit integration of the thermal diffusion equation, the maximum allowable time step can be defined as

$$\Delta t_d = 0.5 \frac{\rho C h^2}{k_{max}}. \quad (19)$$

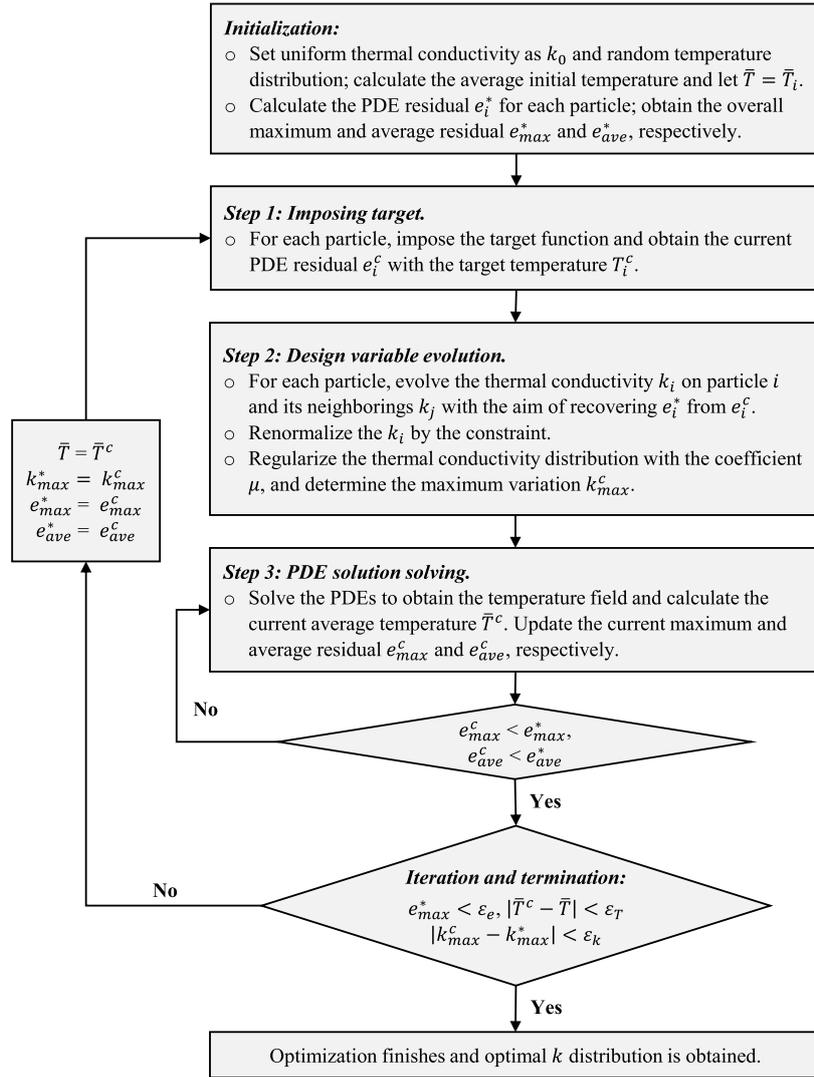


Fig. 2. The flowchart of the target-driven optimization procedure.

Since the implicit scheme is employed here for obtaining the steady solution of the Eq. (6), the time step size is chosen as a large value of  $10\Delta t_d$  without considering the temporal accuracy.

## 4. Target-driven optimization

### 4.1. Method overview

The target-driven PDE-constrained optimization method proposed here is based on the principle of residual recovery. The primary objective is to solve the PDEs while progressively imposing the target function directly defined by the state variable. As the latter part can potentially lead to modifications or even an increase in the PDE residuals, to address this issue, the design variable undergoes an evolution to recover the original residual, before the PDE solving continues. This process is repeated iteratively until the fields of both state and design variables converge and reach steady states. At that point, the optimal distribution of the design variable is obtained while satisfying both the target function and PDE constraints. Note that, the residual recovery approach is a typical all-at-once method since the residual of PDE only converges upon completing the optimization process.

Although the approach proposed above can theoretically be applied to general PDE-constrained optimization problems, here we apply it to

address heat-conduction based optimizations. The flowchart of the proposed method is illustrated in Fig. 2 and the detailed steps are given as follows.

### Initialization:

- The thermal domain is populated with inner and dummy particles for implementing the SPH method. The thermal conductivity is initialized as a uniform distribution with  $k_0$ . The initial temperature is randomly assigned, and then, the average initial temperature  $\bar{T}$  is calculated.
- The PDE residual  $e_i^*$ , i.e. the LHS of Eq. (6), is calculated for each particle within the thermal domain. The overall maximum and average residuals, denoted as  $e_{max}^*$  and  $e_{ave}^*$ , respectively, are then determined.

### Step 1: Imposing target.

- The target is locally imposed on each particle with a strength  $\beta$ . In this case, it can be expressed as  $T_i^c = T_i^* - \beta$  based on the Eq. (4). The modified PDE residual  $e_i^c$  on each particle is then updated with the imposed temperature  $T_i^c$ .

**Step 2: Design variables evolution.**

- The thermal conductivity  $k_i$  on each particle and its neighboring particles  $k_j$  are evolved with the aim of recovering the residual  $e_i^c$  from  $e_i^c$ , as detailed in Section 4.2.
- Re-normalize the thermal conductivity  $k_i$  on each particle to fulfill the constraint presented in Eq. (5).
- The thermal conductivity distribution is regularized with the coefficient  $\mu$ , employing the diffusion analogy numerical regularization method, as explained in detail in Section 4.3. The maximum thermal conductivity variation, denoted as  $k_{max}^c$ , is determined by Eq. (24).

**Step 3: PDE solution solving.**

- The PDE solving advances using the updated thermal conductivity values to obtain the intermediate temperature field. Meanwhile, the average temperature for the current state, represented as  $\bar{T}^c$ , is calculated. The maximum and average residual,  $e_{max}^c$  and  $e_{ave}^c$  respectively, are also updated accordingly.
- The PDE solving stops advancing when the current maximum residual  $e_{max}^c$  and average residual  $e_{ave}^c$  are both smaller than their respective values obtained at the last iteration, satisfying the conditions  $e_{max}^c < e_{max}^*$  and  $e_{ave}^c < e_{ave}^*$ .

**Iterations and termination.** After updating the new  $e_{max}^*$  and  $e_{ave}^*$ , the optimization process repeats from Steps 1 to 3 until the maximum residual reaches the specified threshold ( $1 \times 10^{-5}$ ), and the variations of temperature and thermal conductivity become smaller than the thresholds ( $1 \times 10^{-3}$ ). Once these criteria are met, the optimization process is considered complete, and the resulting distribution of  $k$  is deemed optimal.

Note that the present method divides the optimization process into small, easily manageable steps, which simplifies and expedites the optimization process. The magnitude of target strength  $\beta$  is chosen as a small fraction of the average initial temperature, within the range of  $0.5 \sim 1$  in the present study. In addition, the  $\beta$  is adjusted dynamically, increasing by a factor of 1.05 when the average temperature is lower than in the previous optimize iteration and decreasing with a decayed factor of 0.8 when the temperature exceeds that of the previous iteration. Actually, decreasing  $\beta$  is quite important for effective convergence in the late stages of the optimization. Furthermore, since the convexity of these optimization problems are not able to be established, the current method, similar to other general optimization approaches, does not guarantee the global optimal solution.

**4.2. Evolution of design variable**

In Step 3, the residual  $e_i^*$  for particle  $i$  in the PDE is calculated as

$$e_i^* = \sum_j (k_i + k_j) \frac{T_{ij}}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i. \tag{20}$$

Once the target is imposed on this particle, the PDE residual deviates from its original value and will be recovered by modifying the design variable  $k$  on particle  $i$  and its neighboring particles  $j$ . This process can be represented by the pseudo-time evolution of following equation

$$\frac{dk_i}{d\tau} = \sum_j (T_i^c - T_j) \left( k_i^{m+1} + k_j^{m+1} \right) \frac{1}{r_{ij}} \nabla_i W_{ij} V_j + \dot{Q}_i + e_i^*. \tag{21}$$

Here,  $k_i^{m+1} + k_j^{m+1} = k_i^m + dk_i + k_j^m + dk_j$ , where  $m$  is the previous time step and  $dk_i$  and  $dk_j$  represent increments after the new time step. The implicit splitting operator introduced in Section 3.2 is utilized. Similar to the Eq. (16), a linear system is formed with respect to  $(dk_1, dk_2, \dots, dk_{N-1}, dk_N)^T$ . Note that, the pseudo-time derivative on

the LHS is essential for the stable evolution of  $k$ . If this term is omitted, the diagonal entries of the matrix for the linear system become

$$\left( \eta_1 \sum_j^n B_{1j}, \eta_2 \sum_j^n B_{2j}, \dots, \eta_N \sum_j^n B_{Nj} \right)^T. \tag{22}$$

It is observed that the magnitudes of diagonal entries in the matrix can be significantly smaller than those of non-diagonal entries, potentially leading to numerical instability [55]. On the contrast, when the pseudo-time derivative term is included, the linear system transforms into

$$\left( \eta_1 \sum_j^n (B_{1j} - 1), \eta_2 \sum_j^n (B_{2j} - 1), \dots, \eta_N \sum_j^n (B_{Nj} - 1) \right)^T, \tag{23}$$

whose diagonal entries becomes dominant, and therefore stabilize the evolution of the design variables. In addition, since  $k$  is a material property and should be non-negative, it is clipped at a lower bound of 0.0001 during each iteration.

**4.3. Numerical regularization**

After the evolution of the design variable, it is necessary to apply numerical regularization, serving two essential purposes. One is that, as previously mentioned in the Ref. [15], the regularization plays a critical role in maintaining numerical stability and obtaining a smooth solution. Secondly, it helps prevent over-fitting and avoids finding trivial local optima only. In this study, we introduce a diffusion analogy approach for regularizing the distribution of the design variable, i.e. the thermal conductivity  $k$  is treated as the variable again in the pseudo-time SPH discretized diffusion equation, given as

$$\frac{dk_i}{d\tau} = 2 \sum_j \mu \frac{k_{ij}}{r_{ij}} \nabla_i W_{ij} V_j, \tag{24}$$

where  $k_{ij} = k_i - k_j$  and  $\mu$  is the artificial diffusion coefficient used to control the rate of regularization. We choose  $\mu$  to be general in the range  $1 \sim 2$  according to the target strength. The coefficient also undergoes a similar dynamical adjustment strategy as the target strength because smaller  $\beta$  usually require less regularization to achieve a smooth field. Note that the pseudo-time derivative term is also used in Eq. (24) to ensure that the diagonal is dominant.

**5. Results and discussion**

In this section, we present a set of example problems to validate the effectiveness of the proposed target-driven PDE-constrained optimization method (referred to as the TD) for optimizing the thermal conductivity distribution within a heat conduction domain.

**5.1. The 2/10 heat sinks with uniform internal heat source**

The first set of problems involves a square thermal domain measuring 1m on each side, with a uniform internal heat source. Two heat sinks, each covering 20% of the side length, are symmetrically positioned at the center of opposite boundaries, maintaining the constant temperature. The remaining boundaries are set as adiabatic. Various scenarios are considered, including different heat source intensities, initial thermal conductivity values, and cases with either identical or non-identical sink temperatures. As shown in Fig. 3, the detailed setups for these problems are illustrated as follows:

- Problems 1 and 2 feature a uniform heat source with an intensity of 1000W/m<sup>3</sup>, while Problems 3 and 4 have a uniform heat source with an intensity of 2000W/m<sup>3</sup>.
- The initial thermal conductivity, denoted as  $k_0$ , is 1W/(m·K) for Problems 1 and 2, while it is 4W/(m·K) for Problems 3 and 4.

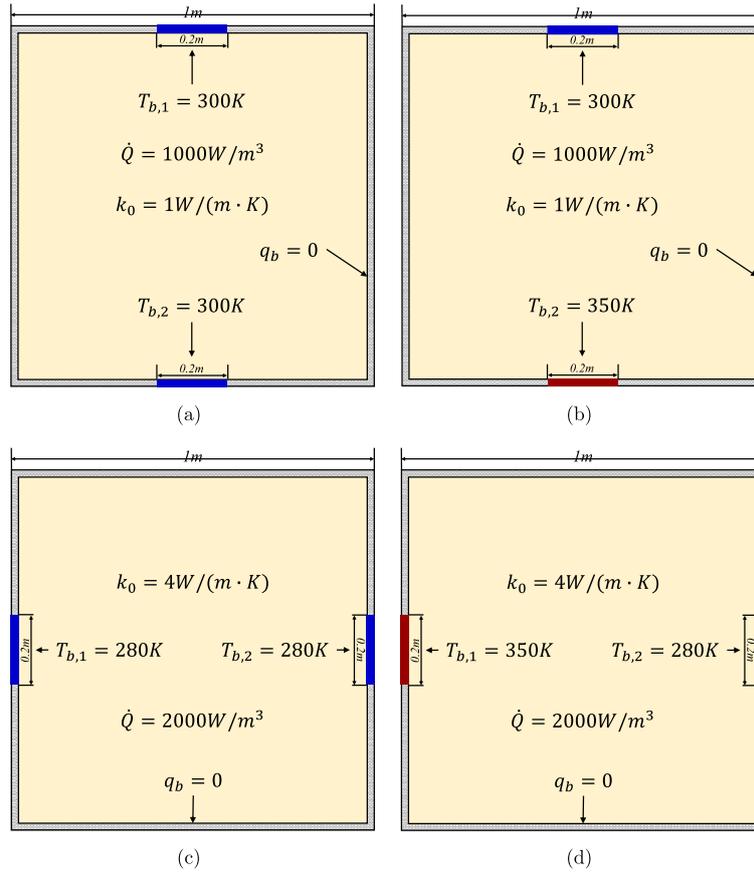


Fig. 3. Illustration of the problem setups. (a) Problem 1: Identical heat sinks; (b) Problem 2: Non-identical heat sinks; (c) Problem 3: Identical heat sinks with higher  $k_0$ ; (d) Problem 4: Non-identical heat sinks with higher  $k_0$ .

- Problems 1 and 3 have identical heat sink temperatures, set at 300K and 280K, respectively. Problems 2 and 4 involve non-identical heat sink temperatures, with one of the heat sinks reaching a higher temperature of 350K.

The thermal domain is discretized with 100 particles on each side, resulting in a total of 10,000 particles. Additionally, four layers of dummy particles are used to enforce boundary conditions. Note that the problems discussed in the following sections share the same configuration for the SPH implementation. Reference solutions for Problems 1 and 2 were obtained using automatic differentiation (AD) and the temperature gradient homogenization (TGH) method, and can be found in Ref. [18]. Reference solutions for Problems 3 and 4 were achieved through adjoint analysis (AA) and the TGH method, and are available in Ref. [15].

The steady temperature distributions with uniform thermal conductivity are presented in Fig. 4. All the obtained distributions are in good agreement with the reference results (see Figs. 4 and 10 of Ref. [18]). In these cases, heat generated by the internal source is conducted throughout the domain and subsequently dissipated at the heat sinks. A more pronounced temperature gradient is observed near the heat sinks, indicating a concentration of heat flux and resulting in higher temperature within the domain. The average temperatures obtained are slightly higher than that in the reference, which may be attributed to variations in resolution and discretization employed.

For these problems, the optimization results obtained by the present method are displayed in Figs. 5 to 8, and the comparisons with previous work are summarized in Tables 1 and 2. Problems 1 and 3, featuring identical heat sinks, result in a symmetrical pattern in the optimized results that align with the reference results. In cases where the heat sinks are identical, the problem becomes self-adjoint, making the TGH

Table 1

Summary of result comparisons for problems with identical sinks.

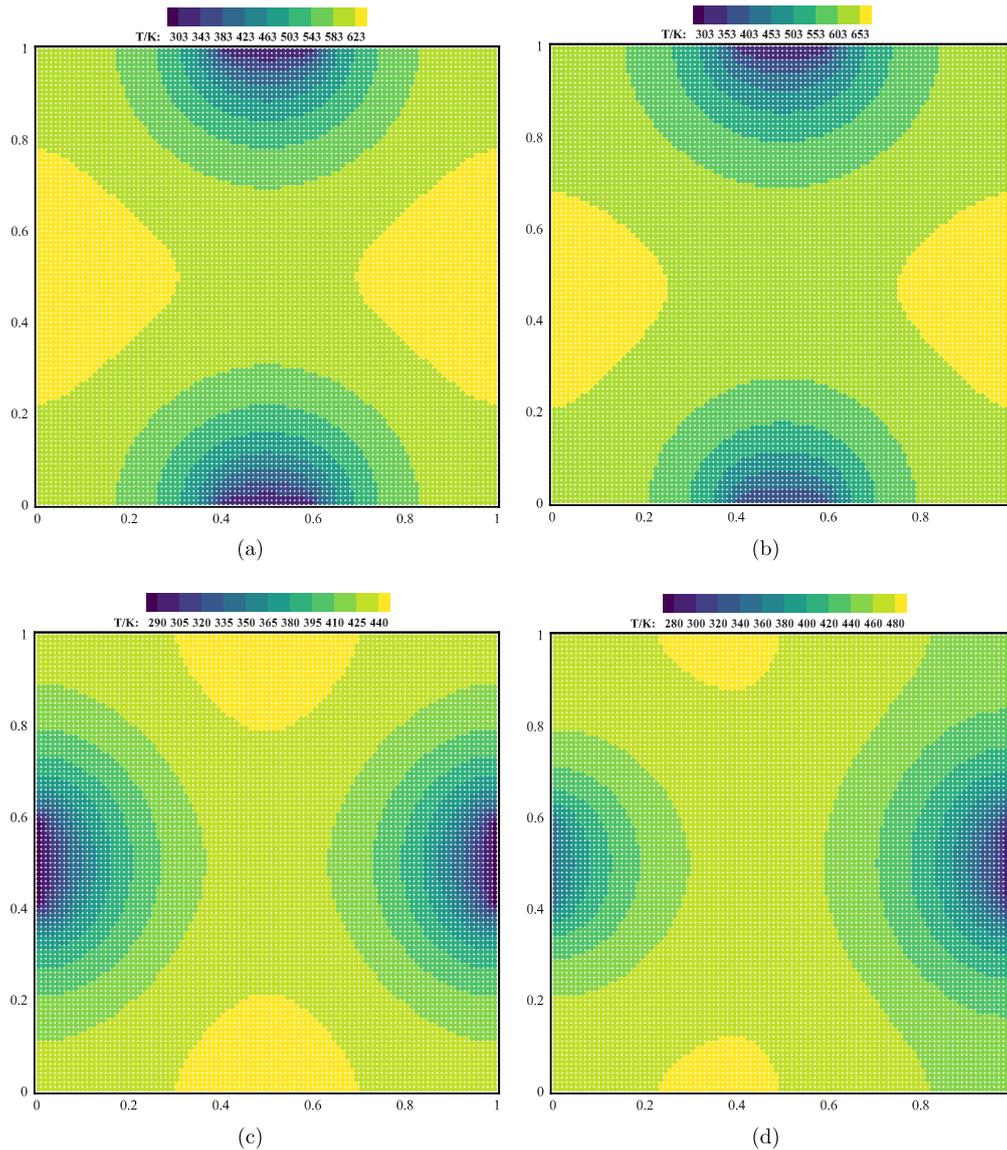
Problem	Method	Original $\bar{T}$ (K)	Optimized $\bar{T}$ (K)	Reduced (%)
1	TD	584.30	413.88	29.17
	AD [18]	582.04	412.73	29.09
	TGH [18]	582.04	413.45	28.97
3	TD method	417.59	339.12	18.79
	AA [15]	414.36	336.98	18.67
	TGH [15]	414.36	336.99	18.67

Table 2

Summary of result comparisons for problems with non-identical sinks.

Problem	Method	Original $\bar{T}$ (K)	Optimized $\bar{T}$ (K)	Reduced (%)
2	TD	609.30	438.79	27.98
	AD [18]	607.04	436.96	28.02
	TGH [18]	607.04	438.20	27.81
4	TD	452.73	371.66	17.91
	AA [15]	449.36	368.61	17.97
	TGH [15]	449.36	371.55	17.32

method equivalent to the AA method when the target is to minimize the global temperature [56]. Therefore, all methods adopt the direct target and exhibit similar optimal performance, although the reference result for the Problem 1 suggests slightly superior outcome using the AD method. Notably, the present method tends to yield a higher reduction in temperature for these two problems. The optimization process homogenized the temperature gradient to balance heat flux through the domain. Optimized temperature distributions in Figs. 5(a) and 7(a)



**Fig. 4.** Temperature distributions with uniform thermal conductivity for different scenarios. (a) Problem 1; (b) Problem 2; (c) Problem 3; (d) Problem 4. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

reveal an overall even temperature gradient, matching with the reference results obtained by other methods (see Fig. 5(b) of Ref. [18] and Fig. 6(a) of Ref. [15]). Furthermore, the optimization process results in a significant increase in thermal conductivity near the heat sinks. The distribution of optimized thermal conductivity in Figs. 5(b) and 7(b) exhibits four distinct peaks prominently located at the edges of the heat sinks, closely matching the reference (see Fig. 5(a) of Ref. [18] and Fig. 6(b) of Ref. [15]). These peaks effectively interact with other boundaries, enhancing the efficient dissipation of generated heat. It's worth noting, however, that the peak thermal conductivity obtained by the present method approximates 14 W/(m·K) and 40 W/(m·K) for Problems 1 and 3, respectively, which are considerable less than the values in the reference (beyond 20 W/(m·K) and 50 W/(m·K)).

The differences in heat sink configurations in Problems 2 and 4 highlight that the AA and TGH methods are no longer equivalent. The TGH method transitions into an indirect target approach, while the direct target methods, TD, AD, and AA, consistently outperform the TGH. The present method exhibits a slightly lower temperature reduction ratio compared to the other two direct target methods. Interestingly, when we decrease the artificial regularization coefficient ( $\mu$ ), it tends

to yield a lower temperature, albeit within certain limits. However, this adjustment results in significantly higher peak values of the thermal conductivity. Therefore, we have chosen the current coefficient value for a balance between optimization performance and the peak value of thermal conductivity. This choice provides greater flexibility in selecting high thermally conductive but electrically isolated materials for cooling electronic devices without sacrificing optimal performance. The optimized temperature distributions in Figs. 6(a) and 8(a) exhibit a generally even gradient, matching with the reference (see Fig. 11(b) of Ref. [18] and Fig. 7(a) of Ref. [15]). Similarly, the optimized thermal conductivity distributions in Figs. 6(b) and 8(b) still feature four peaks around the sinks. However, the heights of opposite peaks are no longer equal due to temperature discrepancies, with larger heights observed near the colder sinks. These features closely align with the reference results (see Fig. 11(a) of Ref. [18] and Fig. 7(b) of Ref. [15]), and notably, the present method is always characterized with lower peaks.

All simulations and optimizations were performed on a computer equipped with 2 Intel(R) Xeon(R) CPU E5-2680 v4 processors. For Problems 1 to 4, it takes approximately 100 seconds to obtain a steady-state temperature field, and detailed information regarding the optimization

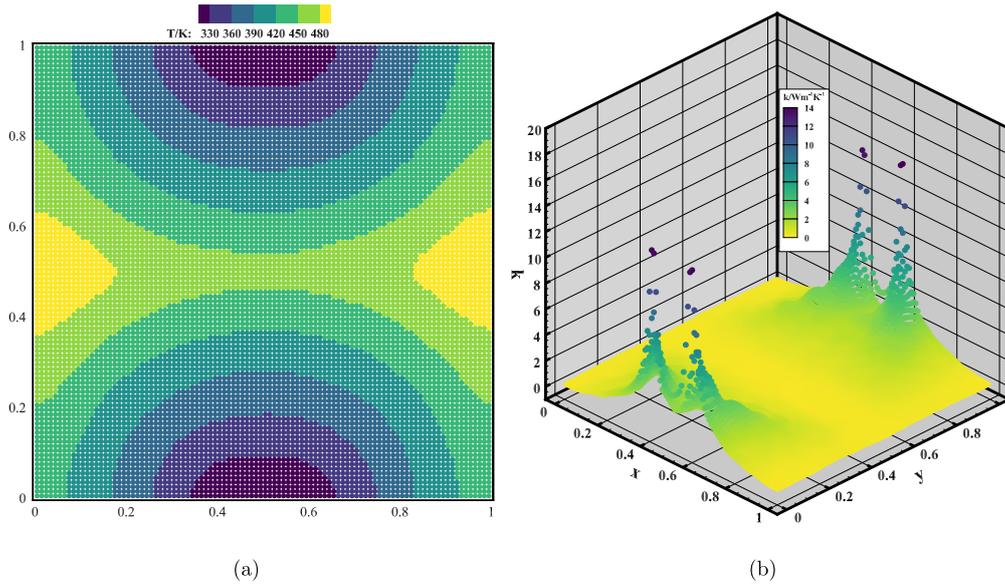


Fig. 5. Present optimized results of the Problem 1. (a) Temperature; (b) Thermal conductivity.

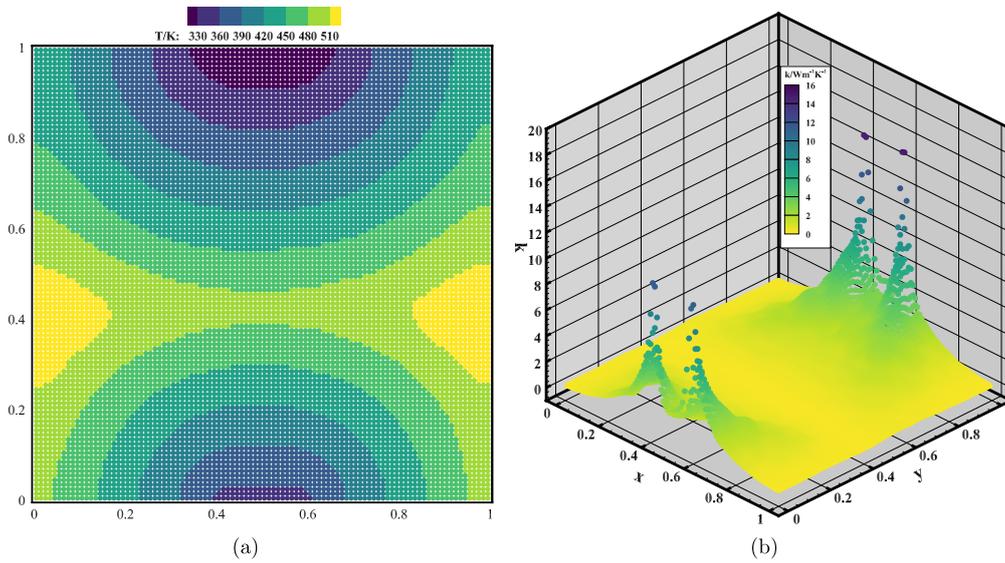


Fig. 6. Present optimized results of the Problem 2. (a) Temperature; (b) Thermal conductivity.

Table 3  
Summary of the optimization process for Problems 1 to 4.

Problem	Steady time (s)	Optimized iteration			Ratio
		Loop	Sub-step	Time (s)	
1	102.8	108	160549	264.3	2.6
2	100.6	94	160541	265.4	2.6
3	110.2	106	210906	341.1	3.1
4	122.1	104	169990	279.5	2.3

duration is presented in Table 3. Note that, with the same convergence criterion, although the actual optimization time for each run may vary and depend on the selected parameters, the shown results suggest that the present method is quite efficient, as it only takes a few times the computation cost of obtaining a converged non-optimized solution to achieve the optimized result. However, simulation-based methods, such as AD, AA, and TGH, need to obtain the converged solution in each iteration to provide information for updating the design variables, which

makes it hard to achieve optimized results with just several iterations, increasing the computation cost for optimization.

### 5.2. The 2/10 sinks with Gaussian distributed heat source

Problem 5 closely resembles Problem 4; however, it introduces a non-uniform distribution of the heat source. Four Gaussian heat sources are symmetrically placed within the domain at four coordinates: (0.25, 0.25), (0.25, 0.75), (0.75, 0.25), and (0.75, 0.75), as depicted in Fig. 9(a). The heat sources  $\dot{Q}_i$  are determined by

$$\dot{Q}_i = C_i \exp \left[ -10 \left( (x - x_i)^2 + (y - y_i)^2 \right) \right]. \tag{25}$$

Here  $(x_i, y_i)$  represents the center point of each heat source, and  $C_i = 3000 \text{ W/m}^3$  denotes the intensity. Fig. 9(b) illustrates the cumulative heat source intensity across the entire domain. Reference solutions obtained by the AA and TGH methods are also available in Ref. [15]. The steady temperature distribution with uniform thermal conductivity

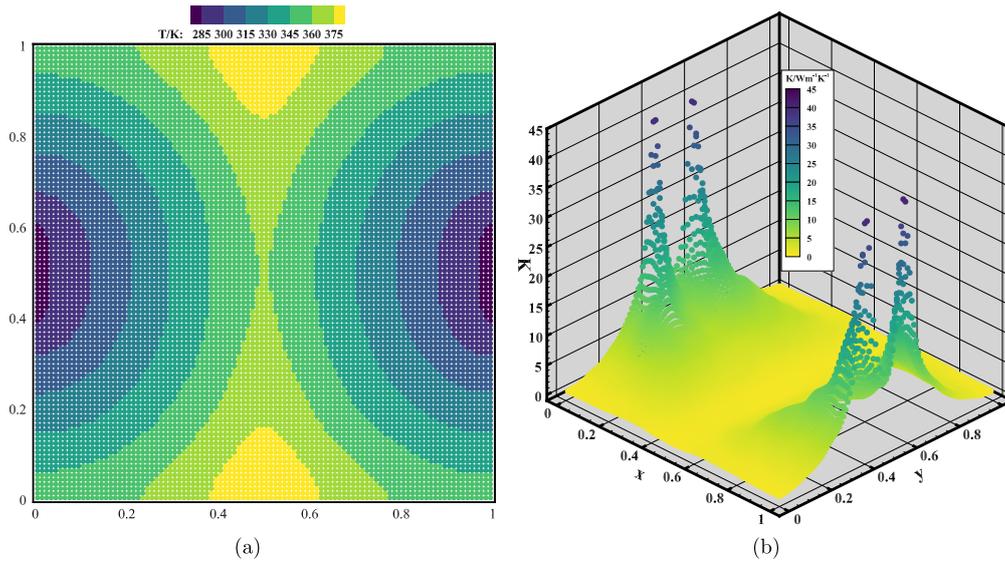


Fig. 7. Present optimized results of the Problem 3. (a) Temperature; (b) Thermal conductivity.

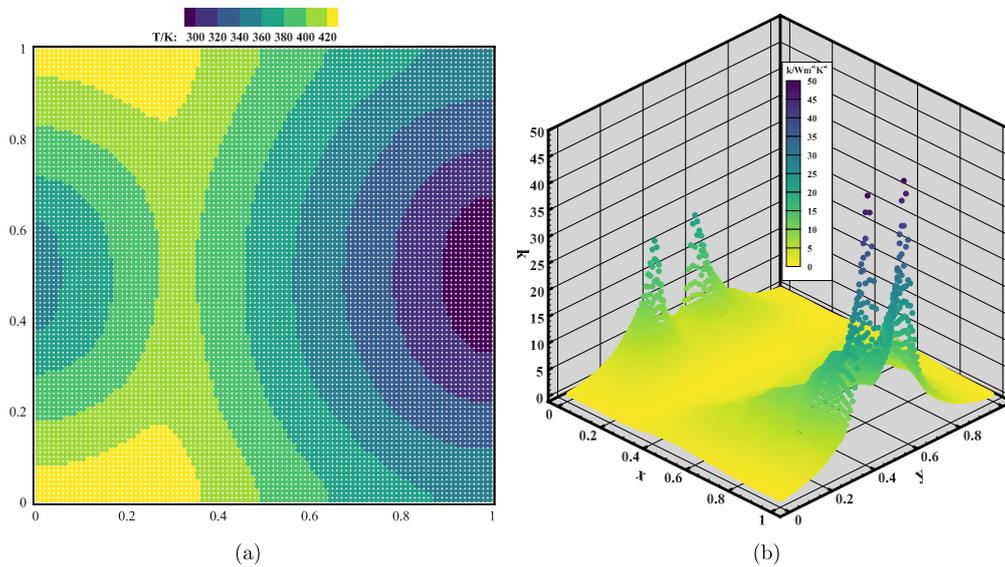


Fig. 8. Present optimized results of the Problem 4. (a) Temperature; (b) Thermal conductivity.

Table 4  
Summary of result comparisons for Problem 5.

Problem	Method	Original $\bar{T}$ (K)	Optimized $\bar{T}$ (K)	Reduced (%)
5	TD	518.55	400.33	22.80
	AA [15]	517.61	417.35	19.37
	TGH [15]	517.61	422.85	18.31

is shown in Fig. 9(c), and the obtained average temperature is 518.55K, which agrees well with the reference.

The current optimization results are presented in Fig. 10, and summarizes the result comparisons in Table 4. Due to the non-identical heat sink configuration, it's evident that the TGH method is no longer equivalent to the AA method, and the reference results demonstrate that AA still outperforms the latter. By employing a direct target, the present method further explores the optimal results and yields more temperature reduction, being aligned with the performance of the AA method, as it is also based on the direct target. The optimized temperature contour shown in Fig. 10(a) reveals a notably lower values, particularly

on the colder heat sink side, indicating an enhanced cooling capacity. The temperature gradient is more evenly distributed, aligning with the reference results (their Fig. 10(a)). Additionally, the optimized distribution in Fig. 10(b) continues to feature the characteristic four peaks similar to the reference (their Fig. 10(b)). However, the present thermal conductivity increases continuously around the heat sink region, as opposed to being concentrated at isolated spots as in the reference. In addition, the simulation time for a steady solution is 112.4 seconds for this case, while the optimization time is 980.7 seconds, again indicating quite good efficiency.

### 5.3. The 1/10 heat sinks with uniform internal heat source

Problem 6 pertains to the utilization of smaller, non-identical heat sinks. As illustrated in Fig. 11(a), two heat sinks, each occupying 10% of the side length, are positioned at the center of the top (350K) and the bottom (300K) boundaries. Reference solutions obtained by the AA and TGH methods are available in Ref. [35]. The temperature distribution under uniform thermal conductivity as shown in Fig. 11(b), is consis-

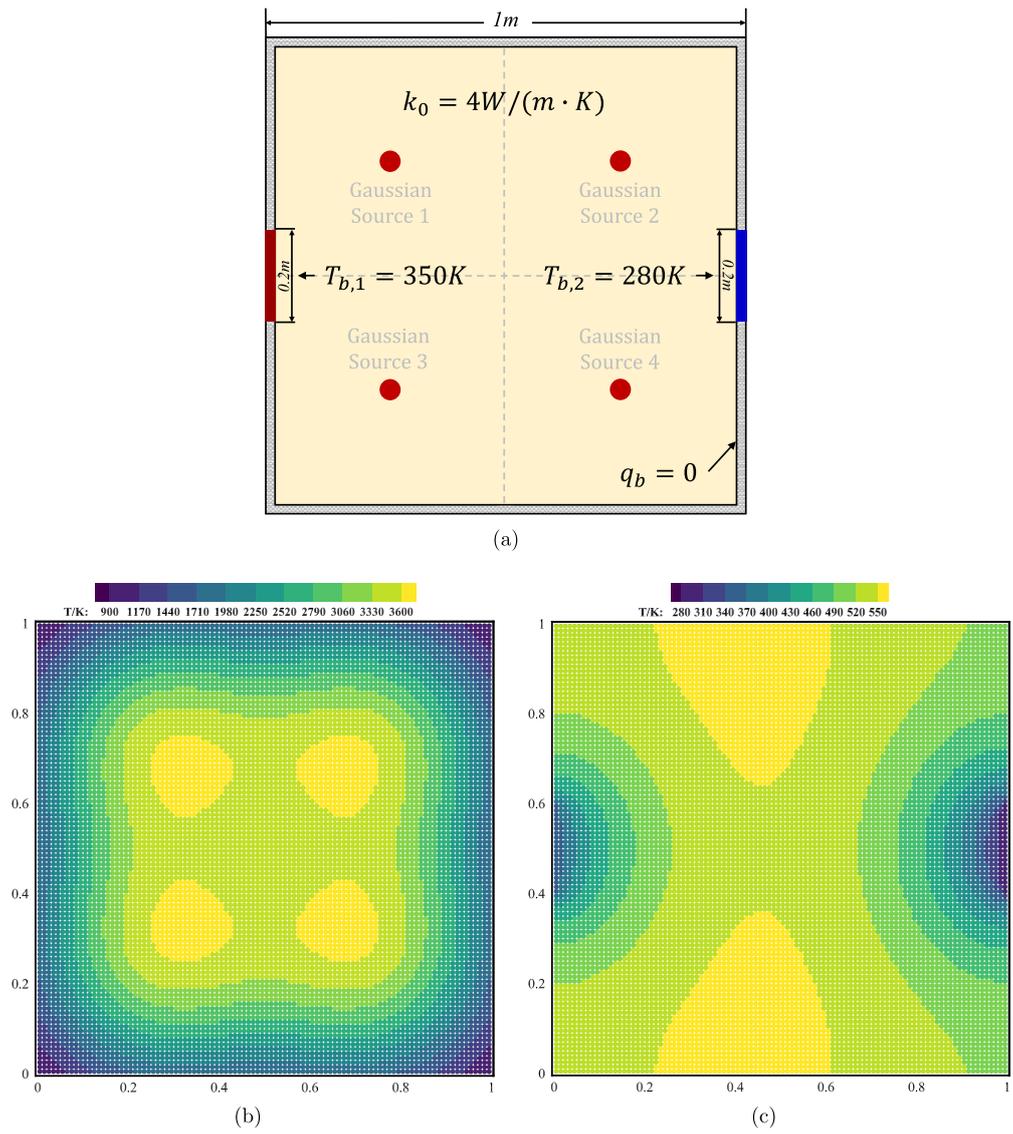


Fig. 9. Problem 5. (a) Schematic depiction of the problem setup; (b) Overall heat source intensity; (c) Temperature distribution under uniform thermal conductivity.

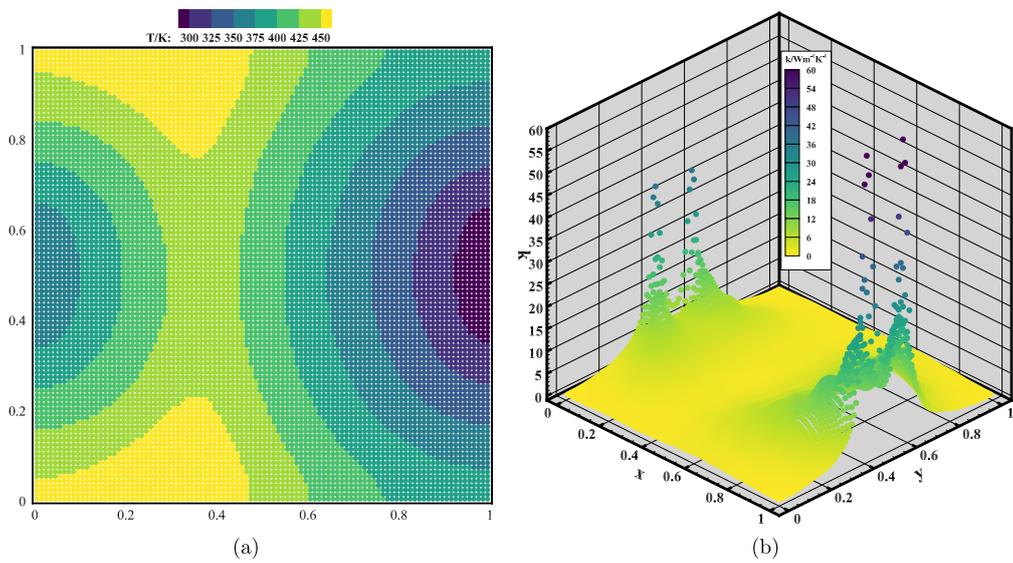


Fig. 10. Present optimized results of the Problem 5. (a) Temperature; (b) Thermal conductivity.

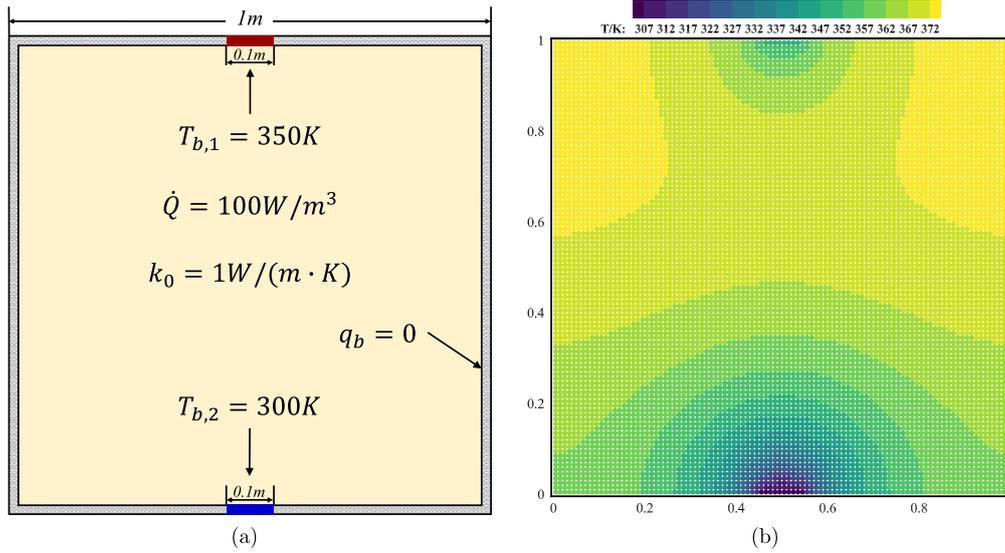


Fig. 11. Problem 6. (a) Schematic depiction of the problem setup. (b) Temperature distribution under uniform thermal conductivity.

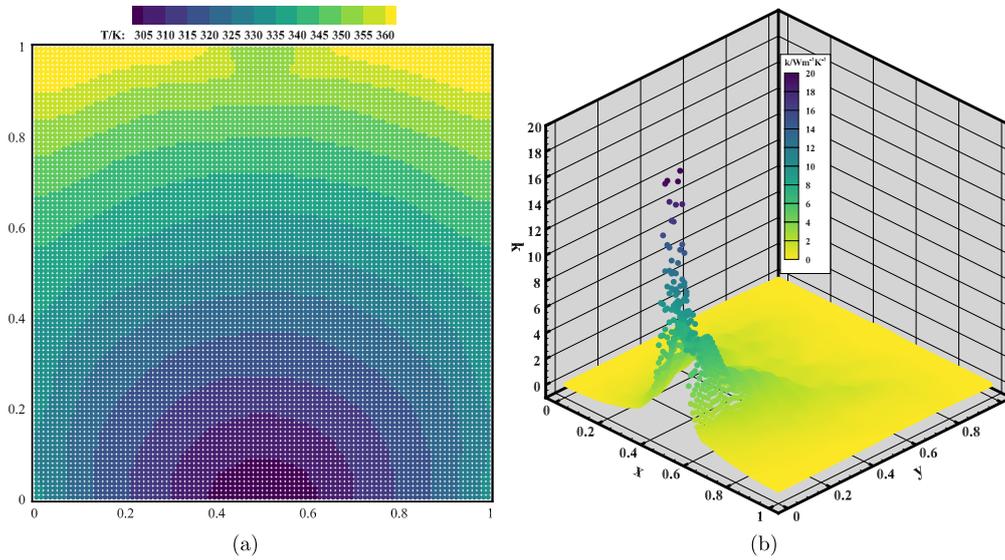


Fig. 12. Present optimized results of the Problem 6. (a) Temperature; (b) Thermal conductivity.

Table 5  
Summary of results comparison for Problem 6.

Problem	Method	Original $\bar{T}$ (K)	Optimized $\bar{T}$ (K)	Reduced (%)
6	TD	365.1	333.0	8.79
	AA [35]	363.3	331.0	8.89
	TGH [35]	363.3	331.9	8.64

tent with the reference results (their Fig. 6(d)) and the present average temperature (365.1K) matches that of the reference (363.3K) also.

The present optimization results are depicted in Fig. 12, and Table 5 provides a summary of comparisons. It is observed that the present temperature reduction ratio is higher than that achieved by the TGH method, although slightly lower than the result obtained using the AA method. As shown in Fig. 12(a), the present optimized temperature distribution presents a smoother profile compared to that of the reference (their Fig. 6(b)), especially in the vicinity of the high-temperature sink. From the present thermal conductivity distribution in Fig. 12(b), the most notable feature is the single peak, in contrast to four peaks obtained in the reference result (their Fig. 7(b)), which refines the mesh

around the small heat sinks. Note that, due to the introduction of regularization, the present highest value of thermal conductivity is only about 20W/(m·K), which is significantly lower than the reference result of about 200W/(m·K). Moreover, the computation time for a steady solution is 137.9 seconds, while the time required for the entire optimization is 890.8 seconds, again showing quite good efficiency.

#### 5.4. The 1/10 sinks with the heat flux heater

Problem 7 explores a thermal domain that features two heat sinks on one side with different operating temperatures, along with a heat flux heater situated on the opposite side. As depicted in Fig. 13(a). A heat flux heater with a magnitude of 2000W/m<sup>2</sup> is positioned at the central of the upper boundary, covering 10% of the side length, and two heat sinks maintained at 300K and 350K, respectively, with the same size are located along the lower boundary. Reference solutions with the target of minimizing temperature on the flux boundary can be found in Ref. [35]. The temperature distribution with uniform thermal conductivity is depicted in Fig. 13(b), and shows good agreement with the reference (their Fig. 10(d)), although the average temperature on the

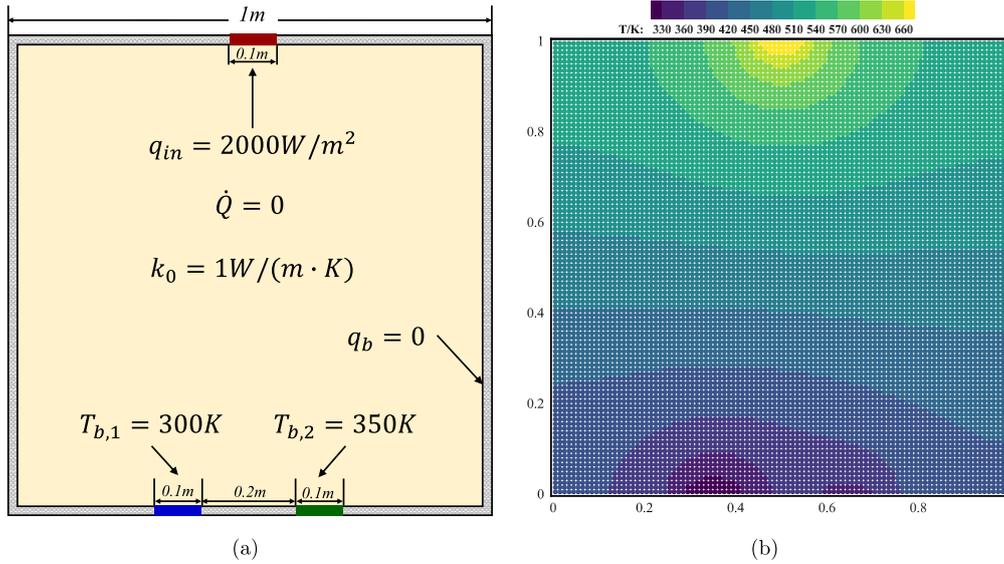


Fig. 13. Problem 7. (a) Schematic depiction of the problem setup. (b) Temperature distribution under uniform thermal conductivity.

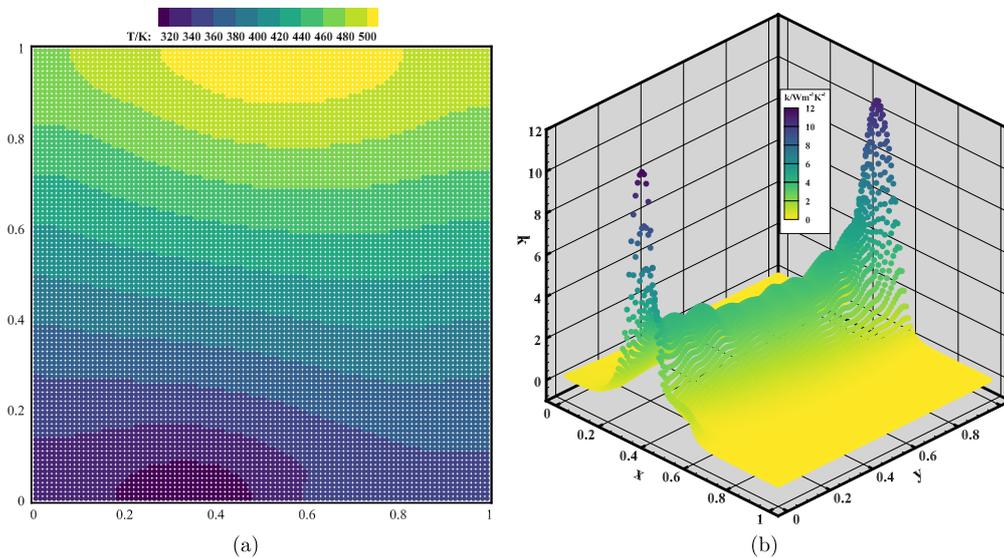


Fig. 14. Present optimized results of the Problem 7. (a) Temperature; (b) Thermal conductivity.

Table 6  
Summary of results comparison for Problem 7.

Problem	Method	Original $\bar{T}$ (K)	Optimized $\bar{T}$ (K)	Reduced (%)
7	TD <sup>1</sup>	470.7	415.3	11.77
	TD <sup>2</sup>	689.5	519.5	24.66
	AA [35]	693.2	501.6	27.64

<sup>1</sup> The average temperature across the entire thermal domain.  
<sup>2</sup> The average temperature along the heat flux boundary.

flux boundary is slightly lower than that in the reference, possibly due to the mesh refinement near the flux heater in the latter.

The present results, with the objective of minimizing the average temperature of the domain, are shown in Fig. 14, and the comparison of results is summarized in Table 6. The optimized thermal conductivity contributes to an overall reduction in temperature. Note that, even though the present objective is different from that of the reference, the pattern of the optimized thermal conductivity in Fig. 14(b) closely resembles that of the reference (their Fig. 11(b)). It forms a bridge

with high thermal conductivity between the flux heater and the colder sink. Compared to the reference results (their Fig. 10(b)), the present optimized temperature distribution also exhibits a uniform gradient perpendicular to the line connecting the heater and the colder sink, proving the enhanced cooling capacity of the colder sink. However, due to the different objectives, the present result shows a higher averaged temperature on the flux boundary compared to the reference. Again, the computation time for the steady solution is 197.9 seconds, whereas the optimization requires 970.8, still suggesting good efficiency.

## 6. Conclusion and remark

In this paper, a target-driven all-at-once approach for PDE-constrained optimization is introduced, and it is applied to optimize thermal conduction problems. By splitting the optimization iteration into small, easily managed steps and treating both state and design variables in the same way, the need for deriving complex adjoint equations and obtaining converged state solutions at each optimization iteration is eliminated. In addition, the mesh-free, splitting-operator based implicit

SPH method is employed as the underlying numerical method, and a diffusion-analogy regularization approach is developed to ensure the numerical stability. Typical examples of thermal conduction problems demonstrate that the present method is able to achieve quite efficient optimization with the computational cost generally on the same order as obtaining a single converged PDE solution. Furthermore, the present optimal results are comparable to those from the previous work, but with the industrially relevant advantage of lower extreme values. Note that, as the target-driven concept used in the present method is not restricted to specific optimization targets, it may be extended for, such as those on the domain boundary, topology, or other thermal and fluid dynamics applications, which are also our future work focus.

### CRedit authorship contribution statement

**Bo Zhang:** Writing – original draft, Validation, Methodology, Investigation. **Chi Zhang:** Writing – review & editing, Validation, Software, 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.

### Acknowledgements

The first author would like to acknowledge the financial support provided by the China Scholarship Council (No. 202006230071). C. Zhang and X.Y. Hu would like to express their gratitude to Deutsche Forschungsgemeinschaft (DFG) for their sponsorship of this research under grant number DFG HU1527/12-4. The corresponding code of this work is available on GitHub at <https://github.com/Xiangyu-Hu/SPHinXsys>.

### References

- [1] A.L. Moore, L. Shi, Emerging challenges and materials for thermal management of electronics, *Mater. Today* 17 (4) (2014) 163–174.
- [2] M. Almgöbel, A. Bejan, Conduction trees with spacings at the tips, *Int. J. Heat Mass Transf.* 42 (20) (1999) 3739–3756.
- [3] Z. Guo, X. Cheng, Z. Xia, Least dissipation principle of heat transport potential capacity and its application in heat conduction optimization, *Chin. Sci. Bull.* 48 (2003) 406–410.
- [4] A. Da Silva, G. Lorenzini, A. Bejan, Distribution of heat sources in vertical open channels with natural convection, *Int. J. Heat Mass Transf.* 48 (8) (2005) 1462–1469.
- [5] K. Chen, S. Wang, M. Song, Optimization of heat source distribution for two-dimensional heat conduction using bionic method, *Int. J. Heat Mass Transf.* 93 (2016) 108–117.
- [6] H. Feng, L. Chen, Z. Xie, F. Sun, Constructal design for a disc-shaped area based on minimum flow time of a flow system, *Int. J. Heat Mass Transf.* 84 (2015) 433–439.
- [7] I.A. Ghani, N.A.C. Sidik, N. Kamaruzaman, Hydrothermal performance of microchannel heat sink: the effect of channel design, *Int. J. Heat Mass Transf.* 107 (2017) 21–44.
- [8] E.M. Dede, S.N. Joshi, F. Zhou, Topology optimization, additive layer manufacturing, and experimental testing of an air-cooled heat sink, *J. Mech. Des.* 137 (11) (2015).
- [9] T. Zegard, G.H. Paulino, Bridging topology optimization and additive manufacturing, *Struct. Multidiscip. Optim.* 53 (2016) 175–192.
- [10] A. Bejan, Constructal-theory network of conducting paths for cooling a heat generating volume, *Int. J. Heat Mass Transf.* 40 (4) (1997) 799–816.
- [11] J.C. De los Reyes, *Numerical PDE-Constrained Optimization*, Springer, 2015.
- [12] R. Herzog, *Lectures notes algorithms and preconditioning in pde-constrained optimization*, Technical Report, 2010.
- [13] R. Herzog, K. Kunisch, Algorithms for pde-constrained optimization, *GAMM-Mitt.* 33 (2) (2010) 163–176.
- [14] K. Ito, K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, SIAM, 2008.
- [15] T. Zhao, X. Wu, Z.-Y. Guo, Optimal thermal conductivity design for the volume-to-point heat conduction problem based on adjoint analysis, *Case Stud. Therm. Eng.* 40 (2022) 102471.
- [16] L.B. Rall, G.F. Corliss, An introduction to automatic differentiation, in: *Computational Differentiation: Techniques, Applications, and Tools*, vol. 89, 1996, pp. 1–18.
- [17] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: a survey, *Heliyon* 4 (11) (2018).
- [18] M. Song, K. Chen, X. Zhang, Optimization of the volume-to-point heat conduction problem with automatic differentiation based approach, *Int. J. Heat Mass Transf.* 177 (2021) 121552.
- [19] Q. Chen, H. Zhu, N. Pan, Z.-Y. Guo, An alternative criterion in heat transfer optimization, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 467 (2128) (2011) 1012–1028.
- [20] W. Qi, K. Guo, H. Liu, B. Liu, C. Liu, Assessment of two different optimization principles applied in heat conduction, *Sci. Bull.* 60 (23) (2015) 2041–2053.
- [21] X. Chen, T. Zhao, M.-Q. Zhang, Q. Chen, Entropy and entransy in convective heat transfer optimization: a review and perspective, *Int. J. Heat Mass Transf.* 137 (2019) 1191–1220.
- [22] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 2014.
- [23] W. Alt, K. Malanowski, The Lagrange-Newton method for nonlinear optimal control problems, *Comput. Optim. Appl.* 2 (1993) 77–100.
- [24] X. Cheng, Z. Li, Z. Guo, Constructs of highly effective heat transport paths by bionic optimization, *Sci. China, Ser. E, Technol. Sci.* 46 (2003) 296–302.
- [25] X. Cheng, X. Xu, X. Liang, Homogenization of temperature field and temperature gradient field, *Sci. China, Ser. E, Technol. Sci.* 52 (2009) 2937–2942.
- [26] Z.-Y. Guo, H.-Y. Zhu, X.-G. Liang, Entransy—a physical quantity describing heat transfer ability, *Int. J. Heat Mass Transf.* 50 (13–14) (2007) 2545–2556.
- [27] A. Bejan, *Entropy Generation Through Heat and Fluid Flow*, 1983.
- [28] A. Bejan, Entropy generation minimization: the new thermodynamics of finite-size devices and finite-time processes, *J. Appl. Phys.* 79 (3) (1996) 1191–1218.
- [29] W. Du, P. Wang, L. Song, L. Cheng, Optimization of volume to point conduction problem based on a novel thermal conductivity discretization algorithm, *Chin. J. Chem. Eng.* 23 (7) (2015) 1161–1168.
- [30] R. Wang, Z. Xie, Y. Yin, L. Chen, Constructal design of elliptical cylinders with heat generating for entropy generation minimization, *Entropy* 22 (6) (2020) 651.
- [31] Q. Chen, X.-G. Liang, Z.-Y. Guo, Entransy theory for the optimization of heat transfer—a review and update, *Int. J. Heat Mass Transf.* 63 (2013) 65–81.
- [32] T. Zhao, Y.-C. Hua, Z.-Y. Guo, Irreversibility evaluation for transport processes revisited, *Int. J. Heat Mass Transf.* 189 (2022) 122699.
- [33] T. Zhao, D. Liu, Q. Chen, A collaborative optimization method for heat transfer systems based on the heat current method and entransy dissipation extremum principle, *Appl. Therm. Eng.* 146 (2019) 635–647.
- [34] S. Wu, K. Zhang, G. Song, J. Zhu, B. Yao, F. Li, Study on the performance of a miniscale channel heat sink with y-shaped unit channels based on entransy analysis, *Appl. Therm. Eng.* 209 (2022) 118295.
- [35] Z.-X. Tong, M.-J. Li, J.-J. Yan, W.-Q. Tao, Optimizing thermal conductivity distribution for heat conduction problems with different optimization objectives, *Int. J. Heat Mass Transf.* 119 (2018) 343–354.
- [36] J. Zhang, X. Wu, M. Song, K. Chen, An effective method for hot spot temperature optimization in heat conduction problem, *Appl. Therm. Eng.* (2023) 120325.
- [37] Z.-Z. Xia, Z.-X. Li, Z. Guo, Heat conduction optimization: high conductivity constructs based on the principle of biological evolution, in: *International Heat Transfer Conference Digital Library*, Begel House Inc., 2002.
- [38] Z.-Z. Xia, X.-G. Cheng, Z.-X. Li, Z.-Y. Guo, Bionic optimization of heat transport paths for heat conduction problems, *J. Enhanc. Heat Transf.* 11 (2) (2004).
- [39] R. Boichot, L. Luo, Y. Fan, Tree-network structure generation for heat conduction by cellular automaton, *Energy Convers. Manag.* 50 (2) (2009) 376–386.
- [40] R. Boichot, L. Luo, A simple cellular automaton algorithm to optimise heat transfer in complex configurations, *Int. J. Exergy* 7 (1) (2010) 51–64.
- [41] X. Xu, X. Liang, J. Ren, Optimization of heat conduction using combinatorial optimization algorithms, *Int. J. Heat Mass Transf.* 50 (9–10) (2007) 1675–1682.
- [42] T. Dbouk, A review about the engineering design of optimal heat transfer systems using topology optimization, *Appl. Therm. Eng.* 112 (2017) 841–854.
- [43] F.H. Burger, J. Dirker, J.P. Meyer, Three-dimensional conductive heat transfer topology optimisation in a cubic domain for the volume-to-surface problem, *Int. J. Heat Mass Transf.* 67 (2013) 214–224.
- [44] M.C.E. Manuel, P.T. Lin, Design explorations of heat conductive pathways, *Int. J. Heat Mass Transf.* 104 (2017) 835–851.
- [45] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (1977) 1013–1024.
- [46] 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.
- [47] V. Vishwakarma, A.K. Das, P. Das, Steady state conduction through 2d irregular bodies by smoothed particle hydrodynamics, *Int. J. Heat Mass Transf.* 54 (1–3) (2011) 314–325.
- [48] F. Garoosi, A. Shakibaenia, Numerical simulation of entropy generation due to natural convection heat transfer using kernel derivative-free (kdf) incompressible smoothed particle hydrodynamics (isph) model, *Int. J. Heat Mass Transf.* 150 (2020) 119377.

- [49] 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.
- [50] E.M. Ryan, A.M. Tartakovsky, C. Amon, A novel method for modeling Neumann and Robin boundary conditions in smoothed particle hydrodynamics, *Comput. Phys. Commun.* 181 (12) (2010) 2008–2023.
- [51] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (3) (1968) 506–517.
- [52] K. Nguyen, A. Caboussat, D. Dabdub, Mass conservative, positive definite integrator for atmospheric chemical dynamics, *Atmos. Environ.* 43 (40) (2009) 6287–6295.
- [53] C.M. Bishop, N.M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, Springer, 2006.
- [54] Y. Zhu, C. Zhang, X. Hu, A dynamic relaxation method with operator splitting and random-choice strategy for sph, *J. Comput. Phys.* 458 (2022) 111105.
- [55] R. Bagnara, A unified proof for the convergence of Jacobi and Gauss–Seidel methods, *SIAM Rev.* 37 (1) (1995) 93–97.
- [56] J. Alexandersen, O. Sigmund, Revisiting the optimal thickness profile of cooling fins: a one-dimensional analytical study using optimality conditions, in: *2021 20th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (iTherm)*, IEEE, 2021, pp. 24–30.

## A.3 Paper III

Bo Zhang, Chi Zhang, Xiangyu Hu

### **Automated regression test method for scientific computing libraries: Illustration with SPHinXsys**

In *Journal of Hydrodynamics*, Volume 36(3), 2024, pp. 466–478, DOI: <https://doi.org/10.1007/s42241-024-0042-6>.

Copyright © 2024 Springer Nature. 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.





## Automated regression test method for scientific computing libraries: Illustration with SPHinXsys

Bo Zhang, Chi Zhang, Xiangyu Hu\*

*TUM School of Engineering and Design, Technical University of Munich, Munich, Germany*

(Received May 27, 2024, Revised July 3, 2024, Accepted July 4, 2024, Published online July 26, 2024)

©China Ship Scientific Research Center 2024

**Abstract:** Scientific computing libraries, whether in-house or open-source, have witnessed enormous progress in both engineering and scientific research. Therefore, it is important to ensure that modifications to the source code, prompted by bug fixing or new feature development, do not compromise the accuracy and functionality that have been already validated and verified. This paper introduces a method for establishing and implementing an automatic regression test environment, using the open-source multi-physics library SPHinXsys as an illustrative example. Initially, a reference database for each benchmark test is generated from observed data across multiple executions. This comprehensive database encapsulates the maximum variation range of metrics for different strategies, including the time-averaged, ensemble-averaged, and dynamic time warping methods. It accounts for uncertainties arising from parallel computing, particle relaxation, physical instabilities, and more. Subsequently, new results obtained after source code modifications undergo testing based on a curve-similarity comparison against the reference database. Whenever the source code is updated, the regression test is automatically executed for all test cases, providing a comprehensive assessment of the validity of the current results. This regression test environment has been successfully implemented in all dynamic test cases within SPHinXsys, including fluid dynamics, solid mechanics, fluid-structure interaction, thermal and mass diffusion, reaction-diffusion, and their multi-physics couplings, and demonstrates robust capabilities in testing different problems. It is noted that while the current test environment is built and implemented for a particular scientific computing library, its underlying principles are generic and can be easily adapted for use with other libraries, achieving equal effectiveness.

**Key words:** Scientific computing, open-source library, verification and validation, regression test, automatic test environment, curve similarity comparison, smoothed particle hydrodynamics

### 0. Introduction

The evolution of the computer has elevated scientific computing to an indispensable position across many technologies and industries, ranging from assessing climate changes<sup>[1]</sup>, designing new energy conductors<sup>[2]</sup>, and enhancing our ability to predict and comprehend natural phenomena and engineering systems. Following William's definition of validation and verification in scientific computing<sup>[3]</sup>, it is crucial that scientific computing consistently and accurately represents both the conceptual model and the real world. However, this is a great challenge due to the complex mathematical models and calculations involved, which often necessitate changes to distinct parts of the library and consequently, increase the likelihood of mistakes. Furthermore, developing a

scientific computing library is a continuous process, marked by frequent changes driven by varying requirements and the introduction of new features. It is essential for libraries to consistently yield trustworthy results throughout their continuous evolution, and achieving this requires a robust validation and verification process conducted in tandem with the development and usage of the library<sup>[4]</sup>.

Implementing testing, encompassing unit tests, integration tests, regression tests, system tests, etc., offers concrete validation and verification procedures. While this practice has gained widespread acceptance in IT software, it faces challenges when endeavoring to conduct tests directly using conventional techniques within scientific computing libraries, particularly for certain open-source libraries. One significant challenge arises from the inherent features of scientific computing libraries, making it challenging to establish a reliable test oracle—A mechanism or criterion used to determine whether the program produces the expected output during test case execution<sup>[5-6]</sup>. The elusive nature of defining precise expected outcomes in scientific contexts adds complexity to the validation

---

**Biography:** Bo Zhang (1995-), Male, Ph. D. Candidate,  
E-mail: [bo.zhang.aer@tum.de](mailto:bo.zhang.aer@tum.de)

**Corresponding author:** Xiangyu Hu,  
E-mail: [xiangyu.hu@tum.de](mailto:xiangyu.hu@tum.de)

process. Additionally, another noticeable challenge arises from cultural differences between scientists and the software engineering community<sup>[5]</sup>. Scientific computing libraries are often developed by small groups of scientists who may not be very familiar with established software engineering practices and have not extensively delved into the development process of their developed libraries<sup>[7]</sup> and, therefore, overlook the potential impacts of changes.

Regression testing stands as a crucial method to ensure the output validity of scientific computing libraries under development. This re-testing activity involves executing a test suite with specified inputs and comparing the output with previously stored reference results, especially when modifications occur or new features are added. Through regression testing, developers can ascertain that their changes do not introduce unexpected side effects and that previous functionalities remain verified<sup>[6]</sup>. Given the time-consuming and tedious nature of performing regression tests for all test cases, especially in large-scale software, many automatic regression test techniques have been developed and successfully implemented to address this challenge, including selection<sup>[8-9]</sup>, minimization<sup>[10-11]</sup>, prioritization<sup>[12-13]</sup>, and optimization of test cases within the test suite.

Concerning the implementation of the regression test in scientific computing libraries, the focus is often on achieving rigorous validation and verification for confidence in computational results. To achieve this, different strategies are employed, and these processes should be automated and continuous<sup>[4]</sup>. Lin et al.<sup>[14-15]</sup> leveraged historical data with multiple inputs and their relationships to define a test oracle, conducting the metamorphic tests. Peng et al.<sup>[16-17]</sup> provided insights into their analysis of released unit tests and regression tests for SWMM, a stormwater management model developed by the U.S. Environmental Protection Agency. Their focus was on test coverage, revealing a novel pattern to address oracle problems. Farrell et al.<sup>[4]</sup> established an automated verification test environment for Fluidity-ICOM, an adaptive-mesh fluid dynamics simulation package. Liu et al.<sup>[18]</sup> developed a web-based automated testing environment to validate computational fluid dynamics (CFD) cases related to high-speed aero-propulsive flows. Happ<sup>[19]</sup> devised a set of Linux C-Shell testing scripts for SHAMRC, a 2-D and 3-D finite difference CFD code addressing airblast problems, running regression tests daily.

Despite the aforementioned efforts to implement testing in mesh-based scientific libraries, a testing environment for meshless libraries is notably absent, even as the popularity of meshless computing grows. Compared with mesh-based methods which have fixed topology, this gap can be attributed to the unique

challenges posed by the Lagrangian property of the meshless methods, where dynamic and varying particle distributions in each computation make it difficult to establish standardized testing procedures. Notably, when the source code of SPHinXsys<sup>[20]</sup>, an open-source multi-physics library developed by our team, is updated, we have encountered instances where certain test cases passed the CTest (CMake Test)<sup>[21]</sup>, yet unexpectedly resulted in simulation crashes without any error output. We had initially believed these test cases to be correct since they passed the testing, only to realize later that they were faulty. This issue proved to be troublesome and was the primary motivation for establishing an effective regression test environment.

In this paper, we introduce a methodological framework for constructing an automatic regression test environment in scientific computing libraries. Initially, a verified reference database for each test case is generated by adopting different types of observing data and stored as the reference. Subsequently, the results obtained from the new version of the code undergo testing against the reference database using a curve-similarity-based comparison. The process of updating the source code automatically activates the regression test for each test case and reports the validation of the results. This regression test method has been implemented for all test cases released in SPHinXsys, covering different dynamic features. To the best knowledge of the authors, this work represents pioneering efforts in regression testing for open-source scientific computing libraries based on the meshless method. The work may also attract attention from the broader scientific computing communities, emphasizing the importance of testing, because if the software is meant to do something, then that can and should be tested<sup>[22]</sup>. It is worth noting that the principles presented in this work are versatile and can be easily adopted in other scientific computing libraries.

## 1. Background

In this section, we briefly introduce the features of the SPHinXsys library and then explain the method for obtaining different tested data.

### 1.1 SPHinXsys

As a fully Lagrangian meshless method, smoothed particle hydrodynamics was proposed for astrophysical applications<sup>[23-24]</sup> and has since found extensive use in simulating a wide range of scientific problems. SPHinXsys is an open-source multi-physics and multi-resolution scientific computing library<sup>[20]</sup> based on SPH, designed to address complex industrial and scientific applications. The currently released version

incorporates several key features, including dual-criteria time-stepping, spatio-temporal discretization, position-based Verlet time-stepping scheme, and multi-resolution, etc. This library efficiently models and solves complex systems, encompassing fluid dynamics<sup>[25-26]</sup>, solid mechanics<sup>[27]</sup>, fluid-solid interaction (FSI)<sup>[28]</sup>, thermal and mass diffusion<sup>[29]</sup>, reaction diffusion<sup>[29]</sup>, electromechanics<sup>[29]</sup>, and many others. For quantitative validation, SPHinXsys includes over 100 test cases with available analytical solutions, experimental data, or numerical results from the literature for comparison. Several other open-source scientific libraries based on the SPH method have also been developed and released for public use, contributing significantly to the SPH community. Examples include GPUSPH<sup>[30]</sup>, SPHysics<sup>[31]</sup>, DualSPHysics<sup>[32]</sup>, AQUApushp<sup>[33]</sup>, GADGET-2<sup>[34]</sup>, and GIZMO<sup>[35]</sup>, etc. Many of these applications/libraries, including the SPHinXsys, are still under intensive development. Given the ongoing development in this field, it becomes essential to introduce a regression test environment to ensure consistent development and releases.

### 1.2 Obtain tested data

For most scientific computing problems, the rarity of acquiring a comprehensive database for the whole computational domain is primarily attributed to the sheer magnitude of data involved, posing challenges in computational resources, storage, and processing. However, observing variables of interest at specific, representative locations is often sufficient to establish a reliable reference. These variables of interest, commonly referred to as variation points, exhibit different values in various executions, known as variants<sup>[6]</sup>. As an illustrative example, in CFD simulations, the pressure probed at a fixed position serves as a variation point. The values it takes at a specific physical time in different executions represent variants. By incorporating variation points, variants, and their associated constraints, a variability model can be generated from a series of computing results. This model then can serve as a valuable reference for regression testing in scientific computing.

In SPHinXsys, the observing data, referred to as the variation point, is categorized into two types. The first type encompasses observed quantities at probes positioned within the computational domain. This includes variables like density, pressure, velocity, etc., in fluid problems, and deformation, stress, displacement, etc., in solids, along with other related variables of interest. The second type involves reduced quantities, representing overall variables of interest in the computational domain. Examples include summation, maximum, and minimum values of a variable, such as the total mechanical energy of the field. It is important

to note that both types of quantities not only serve as the data source for regression testing but also play a crucial role in visualization.

## 2. Method

In this section, we begin by presenting an overview of the regression test procedure, followed by detailed explanations of the individual steps and algorithms for three testing strategies.

### 2.1 Overview of the regression test

The underlying principle of the regression test involves comparing the similarity between verified curves, also referred to as time series results, generated from previously validated executions in the reference database, and newly obtained results after the modification of code. A verified curve typically incorporates a tolerance range to account for uncertainties introduced during different executions. For instance, in shared memory parallel programming libraries like the threading building blocks (TBB), the concurrent vector is commonly used to create a sequence container capable of concurrent growth and access. However, results from multiple executions with the same model may not be identical, displaying noticeable or considerable differences, particularly in highly non-linear problems such as fluid dynamics. In addition, the Lagrangian property of the meshless method results in different particle distributions for multiple executions, leading to discrepancies in the results.

In general, the regression test, illustrated by the flow chart in Fig. 1, comprises two main parts: (1) Generating a reference database for each test case. (2) Automatically verifying the new result obtained after code modification using specific strategies.

For a newly added test case, the following steps can be followed to generate a reference database:

Step 1: Execute the test case and verify the current result with experimental, numerical, or analytical data from the literature.

Step 2: Select one or more variables of interest and define their corresponding thresholds.

Step 3: Set up the testing environment by instantiating objects, and introducing methods for the regression test.

Step 4: Choose and initialize metrics from different strategies according to the type of generated curves or observed time series results. Three different strategies, as explained in the following Sections 2.3-2.5, are applied in this work.

Step 5: Execute the test case multiple times and update metrics with different strategies.

Step 6: Continue the iterations until the variations of all metrics converge under the given threshold, at

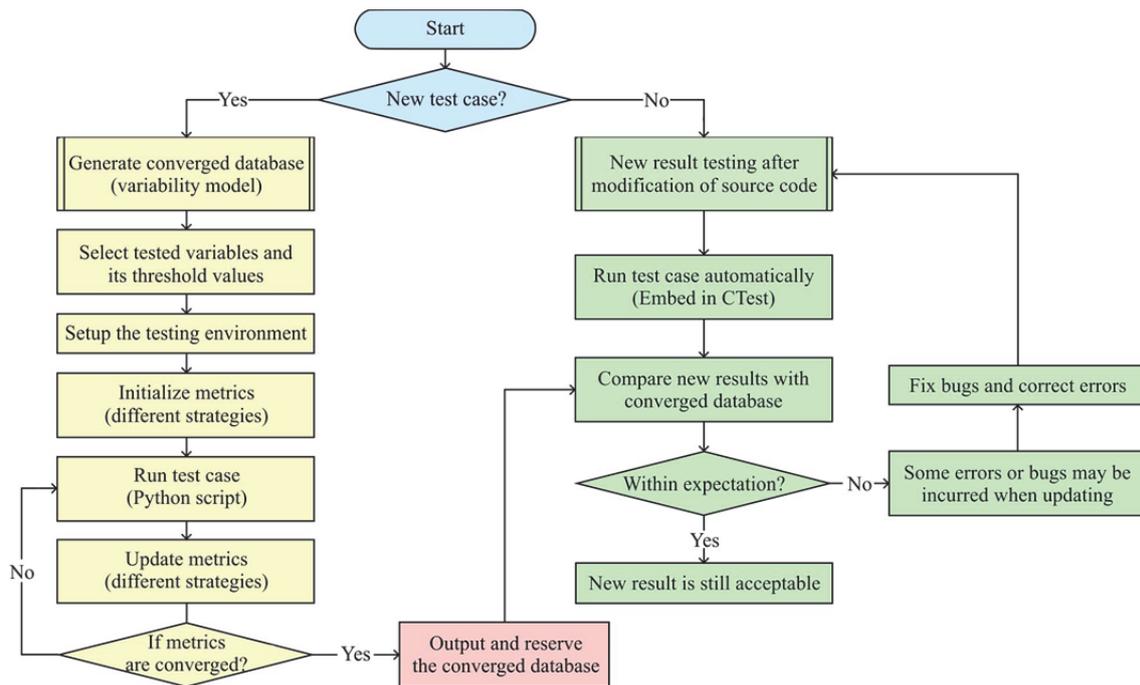


Fig. 1 (Color online) The flow chart for the regression test. The left part is for generating the reference database, and the right one is for testing the new result obtained after code modifications

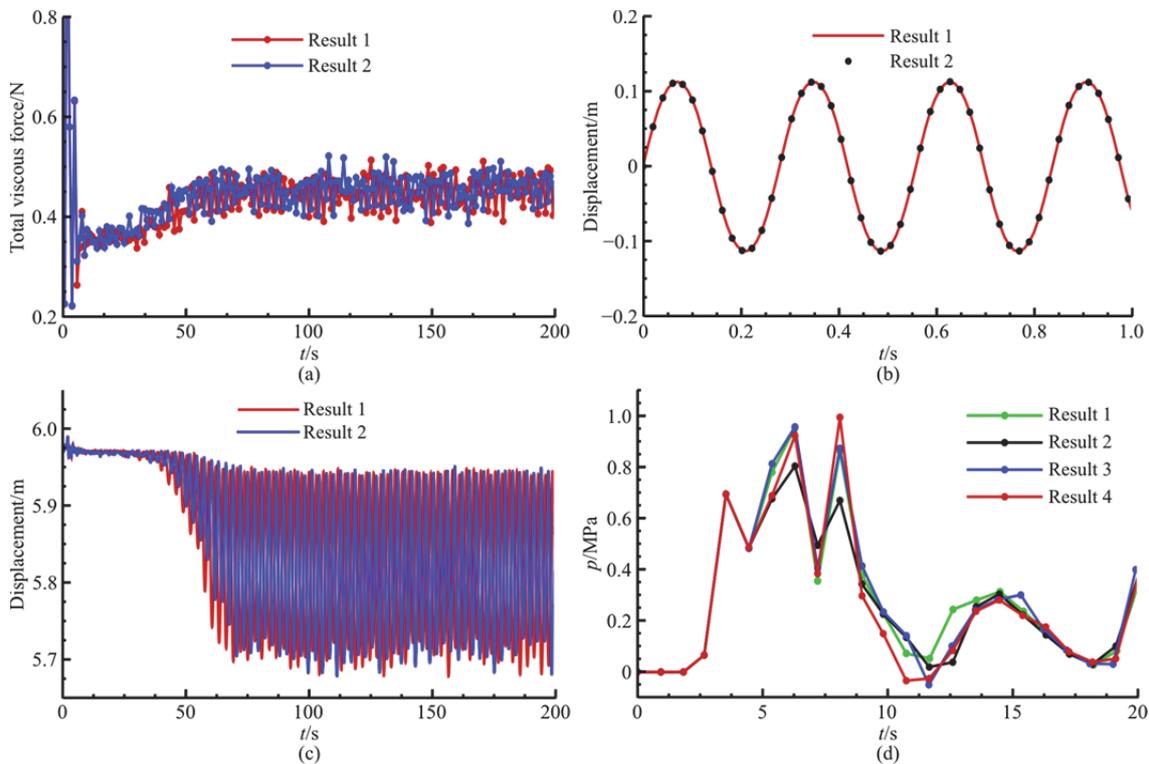


Fig. 2 (Color online) Different types of curves were obtained from time series results and used for the regression test, (a) Type 1: Total viscous force of the flow around a cylinder<sup>[28]</sup>, (b) Type 2: Free end displacement of an oscillating elastic beam<sup>[36]</sup>, (c) Type 3: Flow-induced free end displacement of an elastic beam attached to a cylinder<sup>[28]</sup>, (d) Type 3: Solid wall impacting pressure in dambreak flow<sup>[25-26]</sup>

which point the metrics will be stored as the reference database.

With the reference database in hand, the modified code can be tested using the following steps:

Step 1: Run test cases using CMake Test or other similar testing packages.

Step 2: Compare the newly obtained result with the previously stored one in the reference database based on curve-similarity measures.

Step 3: Check whether the similarity measure is within the given threshold. If it is, the modified code is considered acceptable. Otherwise, investigate the source code for modifications or bugs, and then rerun the testing until the similarity measure is deemed acceptable.

## 2.2 Curve classification and testing strategies

Different strategies should be employed to compare curve similarity based on the types of curves under consideration. For the typical dynamic problems involved in this work, time series data curves can be broadly classified into three types, each corresponding to a distinct comparison strategy.

The first type, associated with the time-averaged strategy, represents data series that fluctuate around a constant value after reaching a steady state. This type of curve is prevalent in many fluid dynamics problems, such as the observed total viscous force for a fluid-structure interaction (FSI) problem, as illustrated in Fig. 2(a). The second type, aligned with the ensemble-averaged strategy, represents data series exhibiting similar variation patterns for each computation. Such curves are often generated from simple solid dynamics problems, such as the displacement of a given point from the oscillating beam presented in Fig. 2(b). The last type, corresponding to the dynamic time warping strategy, represents data series that may experience rapid and scattered variation patterns or large high-frequency fluctuations. These curves are generally produced in simulations characterized by high non-linear dynamics. Figure 2(c) shows an observed position from an FSI simulation, and Fig. 2(d) shows observing pressure for a dambreak flow, both of which exhibit apparent variations in each execution.

## 2.3 Time-averaged strategy

In the time-averaged strategy, as the system consistently reaches a steady state through the relaxation process, metrics such as the time-averaged mean and variance are employed for comparison and testing purposes.

### 2.3.1 Metrics generation and updating

The generation of the reference database under this strategy involves updating time-averaged mean and variance through multiple executions until their

variations converge. For each updating (e.g., the  $n$ th execution), the mean  $M^n$  and variance  $\sigma^n$  of the obtained result  $x$  from the current execution can be calculated as follows:

$$M^n = \frac{l}{l-k} \sum_{i=k}^l x_i^n, \quad \sigma^n = \frac{l}{l-k} \sum_{i=k}^l (x_i^n - M^n)^2 \quad (1)$$

where  $i$  is the index of a data point,  $l$  is the total number of data points and  $k$  is the index of the starting point of the steady state. The mean  $M^*$  and variance  $\sigma^*$  in the regression test metrics are then updated based on the results from the  $n$ th computations. Specifically,  $M^*$  is updated as

$$M^* = \frac{M^n + M^*(n-1)}{n} \quad (2)$$

Note that, instead of storing all previous means  $n$  times, the summation of the mean is recursively updated as a decaying average of all previous means for increased efficiency. Then,  $\sigma^*$  is updated as

$$\sigma^* = \max(\sigma^n, \sigma^*) \quad (3)$$

indicating that the variance is always updated to the maximum variation range. After the relative difference between the newly updated metrics and the previous ones is smaller than thresholds in several successive executions (usually 4 in practical applications), the  $M^*$  and  $\sigma^*$  are stored as the reference database. It is worth noting that the variation of the metrics in two successive runs being smaller than the threshold is only a hint of convergence. This should happen several times successively to ensure a real stable convergence. Therefore, once such variation is larger than the threshold, the count of the converged successive executions will be reset to zero.

### 2.3.2 Start point searching

As the large oscillatory results in the early stage are nonphysical and cannot accurately reflect the true physical state of interest, it is imperative to exclude this portion of the data. To address this concern, a searching technique is proposed to locate the starting point of the steady state, ensuring a reliable calculation of mean value and variance. Specifically, the search begins from the end of the time series, as the simulation time is always set to be sufficiently long to ensure a steady state. To achieve this,  $n$  pieces with the same time interval are sampled from the entire date set. Starting from the end, two successive pieces are averaged separately and compared. As shown in

Fig. 3, the comparison will proceed until the difference between two averages is larger than the given threshold. The earliest pieces will be considered the starting point of the steady state. The detailed procedure is explained in Algorithm 1.

**Algorithm 1:** Search start point

**Input:** The current result *Array* and its length *l*

**Output:** The index of the start point: *k*

Number of data points in each piece:  $n = l / 20$ ,

**for**  $i = L - 1, i \neq 3 \times n, \dots, i$  **do**

$M_1 = 0, M_2 = 0,$

**for**  $j = i, j \neq i - n, \dots, j$  **do**

$M_1 += Array[j] / n,$

$M_2 += Array[j - 2 \times n] / n,$

**end**

**if**  $|(M_1 - M_2) / (M_1 + M_2)| \geq \text{Threshold}$  **then**

$k = \max(k, i - n),$

*break,*

**end**

**end**

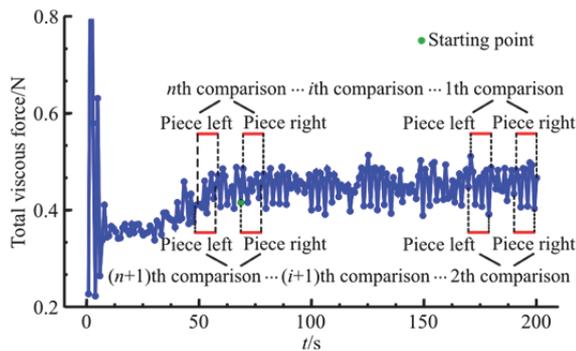


Fig. 3 (Color online) Schematic illustration of the start point searching technique

2.3.3 Regression test

For the regression test, the mean value and variance of the new result are compared with the metrics of the reference database. The correctness of the new result and the compatibility of the modified code with the previous version are determined based on the following conditions:

$$|M - M^*| \leq \alpha M^*, \quad \sigma \leq \sigma^* \tag{4}$$

If these conditions are satisfied, the new result is considered correct, and the modified code is deemed compatible with the previous version. The parameter  $\alpha$  is chosen based on the specific type of dynamics problem. In solid dynamics,  $\alpha$  is set to 0.05, while for fluid dynamics, it is set to 0.1.

2.4 Ensemble-averaged strategy

In the ensemble-averaged strategy, the result

curves obtained from simulation runs typically exhibit similarities within a certain variation range. This range is defined by the metrics of ensemble-averaged mean and variance.

2.4.1 Metrics generation and updating

For the *n*th execution, the metrics for each data point *i* are updated based on the previous values and the new results. The ensemble-averaged mean  $M_i^*$  at a data point *i* is updated as

$$M_i^* = \frac{x_i^n + M_i^{n-1}(n-1)}{n} \tag{5}$$

where  $x_i^n$  is the newly obtained data point,  $M_i^{n-1}$  is the previous mean. Similarly to the time averaged strategy, the new variance  $\sigma_i^*$  is updated as

$$\sigma_i^* = \max(\sigma_i^*, \sigma_i^{n-1}, [0.01(M_{i,\max}^* - M_{i,\min}^*)]^2) \tag{6}$$

where the last term is a secure value introduced to create a variation range and prevent a zero maximum variance for results from different computations. This secure value is determined based on the maximum and minimum values of the local result. The convergence criteria are the same as those in the time-averaged strategy, requiring successive executions with sufficiently small variations of the updated mean and variance to terminate the metrics updating.

2.4.2 Extreme value filter

In some physical issues, there are cases where otherwise smooth and regular results may contain apparent extreme values. These values are often obtained from fast events with insufficient sampling frequency, such as wave-impacting events within generally continuous FSI problems. They may negatively impact the accuracy of the metrics. To address this issue, an extreme value filter is introduced and adopted in some test cases.

As presented in Algorithm 2, the entire dataset is divided into *n* segments, each encompassing a tested data point along with its adjacent data points. Within each data segment, the standard variance of the neighboring data points is compared with the variance between the tested data point and its neighbors. If the latter is four times greater than the former, the tested data point is identified as an extreme value and is subsequently adjusted to the mean of its neighboring data points. Figure 4 presents the filtering process applied to the pressure at a fixed position and the total kinetic energy from an oscillating wave surge converter case<sup>[37]</sup>, which is an ocean engineering application case.

**Algorithm 2:** Extreme value filter

**Input:** The current result *Array* and its length *l*  
**Output:** The smoothed result *Filtered*  
 Length of each data segment:  $n = l / 200$ ,  
**for**  $i = 0, i \neq l, ++i$  **do**  
      $Filtered[i] = Array[i], M_{nhb} = 0, \sigma_{nhb} = 0, \sigma_{test} = 0$   
     **For**  $j = \max(i - n, 0), j \neq \min(i + n, l), ++j$  **do**  
         calculate neighboring points mean  $M_{nhb}$ ,  
     **end**  
     **for**  $j = \max(i - n, 0), j \neq \min(i + n, l), ++j$  **do**  
         calculate neighboring points variance  $\sigma_{nhb}$ ,  
     **end**  
     calculate tested point variance  $\sigma_i = (Array[i] - M_{nhb})^2$ ,  
     **if**  $\sigma_i > 4 \times \sigma_{nhb}$  **then**  
          $Filtered[i] = M_{nhb}$ ,  
     **end**  
**end**

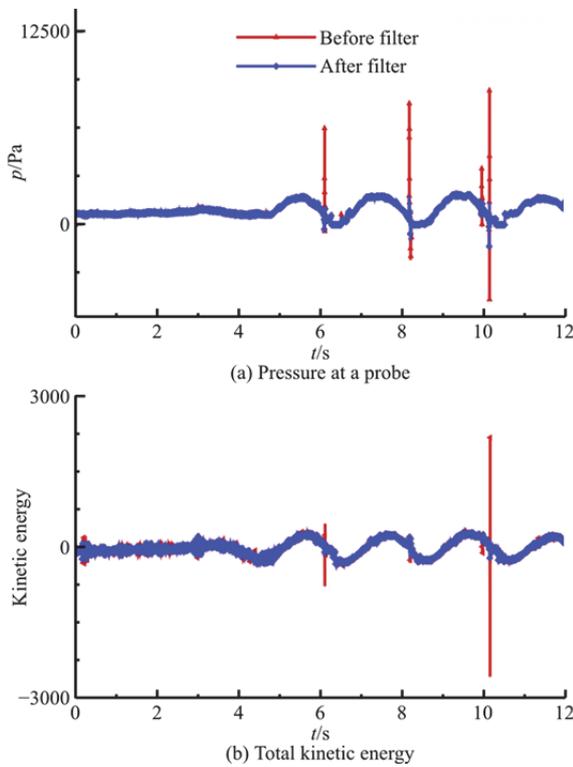


Fig. 4 (Color online) Extreme filtering in the problem oscillating wave surge converter<sup>[37]</sup>

2.4.3 Regression test

With the metrics of the reference database available, the regression test following code modification is conducted for all data points using the following condition

$$\sqrt{(x_i^n - M_i)^2} \leq \sigma_i \tag{7}$$

If there is any data point that does not satisfy this

condition, it indicates that the code modification needs to be reviewed and corrected.

2.5 Dynamic time warping (DTW) strategy

DTW, originally proposed for spoken word recognition<sup>[38-39]</sup>, is a dynamic programming algorithm used to measure the similarity between two sequences with temporal variation by computing the DTW distance. In comparison to the Euclidean distance, DTW distance is more accurate and can handle non-linear distortions, shifts, and scaling in the time dimension. It has found widespread application in various research areas such as sign language recognition<sup>[40]</sup> and time-series clustering<sup>[41]</sup>, among others. Moreover, this algorithm is extensively used in engineering fields involving time-series comparison, including applications in health monitoring and fault diagnosis<sup>[42]</sup>. Due to its generic properties, the DTW strategy may also be employed for curve types that are not classified in Section 2.2.

2.5.1 Calculation of DTW distance

Suppose we have two time series

$$P: p_1, p_2, \dots, p_i, \dots, p_{m-1}, p_m,$$

$$Q: q_1, q_2, \dots, q_j, \dots, q_{n-1}, q_n \tag{8}$$

where *m* and *n* indicate the length of time series *P* and *Q*, while *i* and *j* are data point indices in the time series. The DTW algorithm divides the problem into multiple sub-problems, and each one contributes to the cumulative calculation of the distance<sup>[43]</sup>. The first step is to construct a local distance matrix *d* consisting of *m* × *n* elements, where each element represents the Euclidean distance between two data points in the time series. Then, the warping matrix *D*, seen in Fig. 5, is filled based on the recurrence relation:

$$D(i, j) = d(i, j) + \min \begin{cases} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{cases} \tag{9}$$

Finally, DTW reports the optimal warping path and the DTW distance. The warping path consists of a set of adjacent matrix elements that identify the mapping between two sequences, representing the path that minimizes the overall distance between *P* and *Q*. Each warping path should follow certain rules<sup>[38, 44-45]</sup>: Each index from the first sequence must be matched with one or more indices from the other sequence, and such mapping must be monotonically

increasing. Note that the first index and the last index from the first sequence must be matched with their counterparts from the other sequence correspondingly.

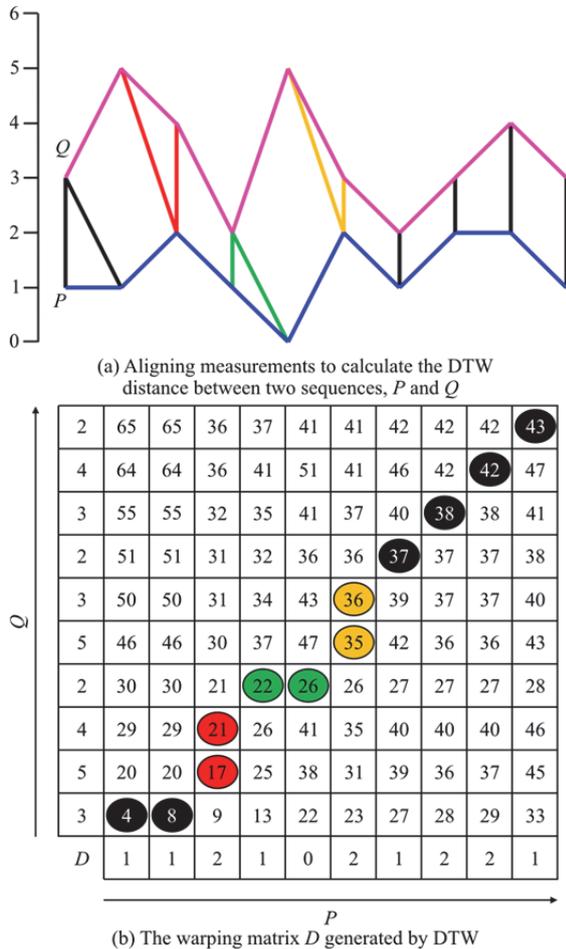


Fig. 5 (Color online) Illustration of the DTW (Labeled cells represent the optimal warping path)

However, DTW can be computationally expensive when searching for global matches, and as a result, many algorithms have been proposed to reduce futile computation<sup>[38, 45-48]</sup>. One effective and simple method for speeding up DTW is to set a warping window (ww)<sup>[38]</sup>. The warping window adds a local constraint that forces the warping path to lie within a band around the diagonal, as shown in Fig. 6, restricting the searching window to a fixed size  $w$ . In the current work, we adopt the window size  $w$  as

$$w = \max(|m - n|, 5) \tag{10}$$

After imposing the constraint, the warping will only occur within the diagonal green areas, and if the optimal path crosses the band, the distance will not be the optimal one.

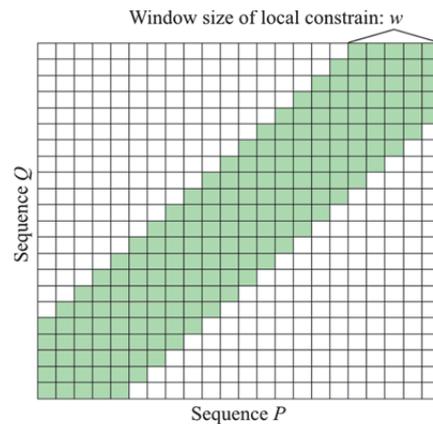


Fig. 6 (Color online) The warping scope is limited by the warping window

### 2.5.2 Metrics generation and updating

The maximum DTW distance is utilized as the regression test metric and is updated after each execution until its variation converges to a certain threshold. With the initial value for the first computation set as  $D_{0,0} = 0$ , the maximum distance for the  $n$ th execution is calculated as

$$D^* = \max(D^*, D_{0,n}, D_{1,n}, \dots, D_{n-2,n}, D_{n-1,n}) \tag{11}$$

where the subscript, e.g.,  $D_{n-2,n}$  denotes the distance between the  $(n - 2)$ th and  $n$ th computational results. Similar to the other two strategies, after the variation  $D^*$  converges to a given threshold in successive several executions,  $D^*$  and several results (usually 3-5 in practical applications) with all data points are stored for the regression test.

```

template <typename VariableType>
class ObservedQuantityRecording :
    public InterpolatingAQuantity< VariableType>
{
    ...
    void writeToFile(size_t iteration_step = 0);
}
    
```

(a) Observation of a variable of interest by ObservedQuantityRecording

```

template <class ReduceMethodType>
class BodyReducedQuantityRecording
{
    ReducedMethodType reduce_method;
    ...
    void writeToFile(size_t iteration_step = 0);
}
    
```

(b) BodyReducedQuantityRecording is to generate reducing variables of interest

Fig. 7 Observing methods in SPHinXsys

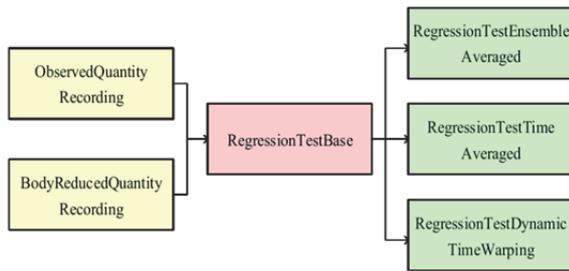


Fig. 8 (Color online) Relationship of classes related to the regression test in SPHinXsys

### 2.5.3 Regression testing

For the regression test, if the DTW distances between the new result after code modification and each result in the reference database satisfy the condition

$$(D_1, D_2, \dots, D_k) \leq D^*, \quad k = 3 - 5 \quad (12)$$

the new result is regarded as acceptable. Otherwise, if any distance exceeds the threshold  $D^*$ , it indicates unexpected behavior, and the code should be checked and corrected.

### 3. Regression test environment

In this section, the process of building an automatic regression test environment is explained, taking SPHinXsys as an example. The test interface is integrated into an SPHinXsys application's case code based on the data observing module. In SPHinXsys, the observing module includes two classes, as depicted in Fig. 7. The observed quantities at probes are generated from ObservedQuantityRecording, and the BodyReducedQuantityRecording could generate the reduced quantities. The above two methods are implemented with the template, allowing the flexible handling of different data types and providing rich data sources for the regression test.

Figure 8 presents the relationship among different regression test methods. RegressionTestBase, the class template, defining commonly used methods in regression tests, inherits from the above observation class. Then, three derived template classes are defined to implement specialized methods, i.e., RegressionTestTimeAveraged, RegressionTestEnsembleAveraged, as well as RegressionTestDynamicTimeWarping. Note that the present structure provides a very flexible combination of test strategies for different variables of interest.

To set up a regression test for a specific test case,

```

1 ***
2 case name: test_2d_dambreak
3 ***
4
5 case_name = "test_2d_dambreak"
6 body_name = "WaterBody"
7 parameter_name = "TotalMechanicalEnergy"
8 body_name_1 = "Fluidobserver"
9 parameter_name_1 = "Pressure"
10
11 number_of_run_times = 0
12 converged = 0
13 sphinxsys = SphinxsysRegressionTest(case_name, body_name, parameter_name)
14 sphinxsys_1 = SphinxsysRegressionTest(case_name, body_name_1, parameter_name_1)
15
16
17 while True:
18     print("Now start a new run.....")
19     sphinxsys.run_case()
20     number_of_run_times += 1
21     converged = sphinxsys.read_dat_file()
22     converged_1 = sphinxsys_1.read_dat_file()
23     print("Please note: This is the", number_of_run_times, "run!")
24     if number_of_run_times <= 200:
25         if (converged == "true") and (converged_1 == "true"):
26             print("The tested parameters of all variables are converged, and the run will stop here!")
27             break
28         elif converged != "true":
29             print("The tested parameters of", sphinxsys.sphinxsys_parameter_name, "are not converged!")
30             continue
31         elif converged_1 != "true":
32             print("The tested parameters of", sphinxsys_1.sphinxsys_parameter_name, "are not converged!")
33             continue
34     else:
35         print("It's too many runs but still not converged, please try again!")
36         break
  
```

Fig. 9 (Color online) Example of the Python script for generating test metrics for a 2d dambreak flow case in SPHinXsys, with two variables of interest

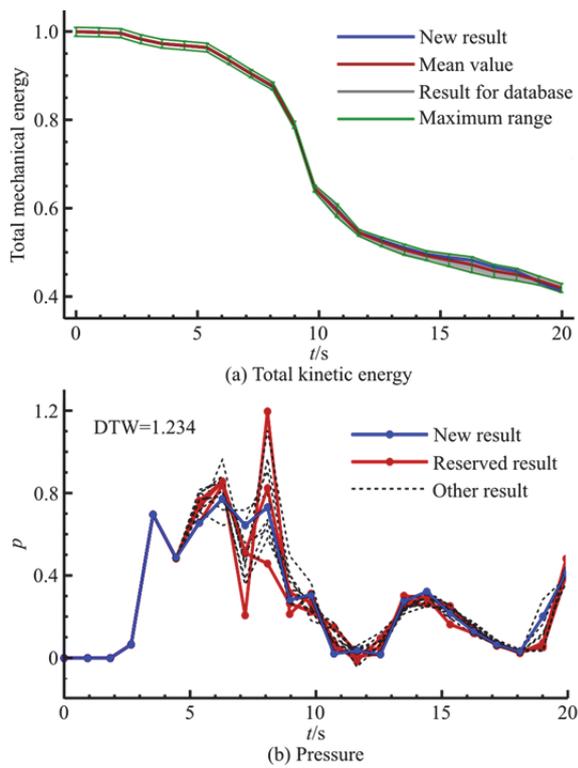


Fig. 10 (Color online) Illustration of the reference database and results for the 2-D dambreak case

it only needs to replace the existing observing class with the regression test class based on the type of

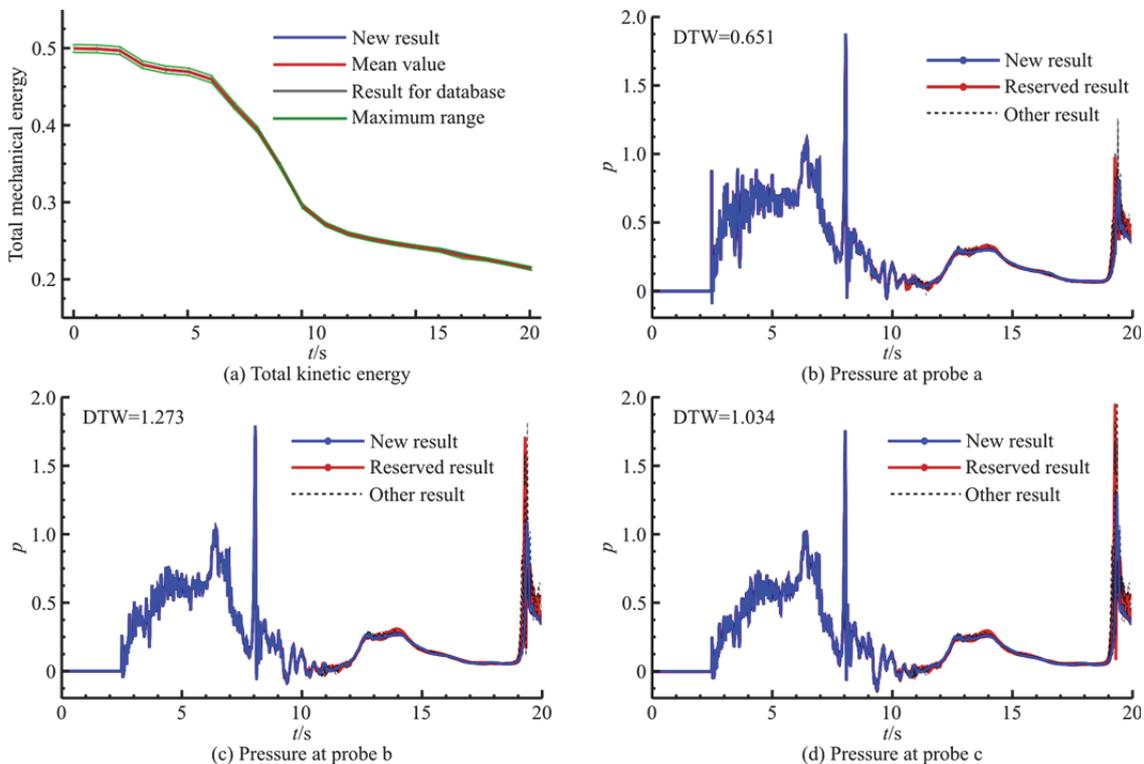


Fig. 11 (color online) Illustration of reference database and results for the 3-D dambreak case

curve that is being observed. Afterward, call the interface `generateDataBase()` to generate a reference database or `testNewResult()` to perform a regression test at the end of the case file.

It is noted that the current method does not disturb the existing code structure. In the SPHinXsys package, a Python script, as exemplified in Fig. 9, is employed to execute a test case multiple times automatically for generating the reference database. This is an automatic process as long as the script is made for the test case, and it is also easy to regenerate the reference database when it is necessary. In SPHinXsys, the regression test is set up for all test cases that simulate real-world problems with the SPH method. Together with Google Test<sup>[49]</sup> for unit test, all tests are integrated using the CTest<sup>[21]</sup>. When merging branches occur, all tests can be triggered automatically.

#### 4. Applications and examples

The reference database obtained for several test cases will be presented here to demonstrate the functionality of the current regression test method.

##### 4.1 Dambreak

The first example involves dambreak flows in both two and three dimensions. The total mechanical energy of the entire domain and pressure at fixed probes have been recorded and validated<sup>[25, 36]</sup>. Therefore, these two observing variables have been

**Table 1 DTW distance in the reference database and for the new results**

DTW distance	2-D	3-D: Probe a	3-D: Probe b	3-D: Probe c
Database	1.234	0.651	1.273	1.034
Testing 1	0.368	0.180	0.229	0.231
Testing 2	0.535	0.177	0.153	0.144
Testing 3	0.522	0.015	0.084	0.111

utilized for the regression test.

According to the classification of curves, the ensemble-averaged strategy is employed for the curve of total mechanical energy, while the DTW strategy is applied to the pressure curve. It is observed that the kinetic energies obtained after code modification, see Figs. 10(a), 11(a), fall within the range of the reference database. The collection of multiple pressure results is given in Figs. 10(b), 11(b)-11(d), where the 3-D case has three pressure monitoring points. After continuously updating the maximum DTW distances for each pair of results, the variation of distances converges, and the final distance is stored, as listed in Table 1. Notably, not all computational results but only several randomly chosen ones are shown here and have been preserved in the reference database. Table 1 indicates that the distance between the newly obtained results and the ones stored in the database are all smaller than the reference distances. Therefore, after performing the regression test on these two variables, new results obtained after code modification are deemed correct, and the new code is considered to be compatible with the old version for the dambreak flow case.

4.2 Oscillating-beam

The second example involves the free-end oscillating elastic beam problem. The detailed setup and validation can be referred to in our previous work<sup>[36]</sup>. The displacement of the beam tip has been recorded and used for the regression test. This variable has two components representing different directions, and it has slight differences for each computation. Therefore, the ensemble-averaged method is adopted for generating the reference database and new result testing. Figure 12 demonstrates the reference database for this case. It is found that the new result of this case (not shown here due to very small, not noticeable visually, differences) after code modifications lies within the range given by the reference database for each data point.

4.3 Fluid-solid interaction

The last example involves a fluid-solid interaction problem on flow-induced vibration. More information and validations can be found in earlier work<sup>[26, 28]</sup>. The total viscous force from the fluid acting on the solid structure was recorded, and it fluctuated around the constant value when the dynamics entered a periodic oscillation state. Thus, the time-averaged

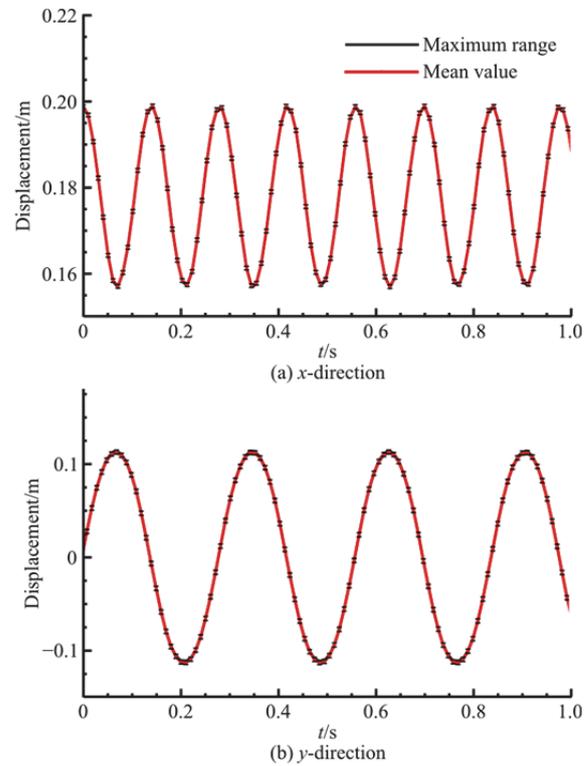


Fig. 12 (Color online) Illustration of the reference database and results for the 2-D oscillating beam case. Displacement along different directions. The red lines represent the ensemble-averaged mean for each data point, and the gray ones give the range of maximum variation

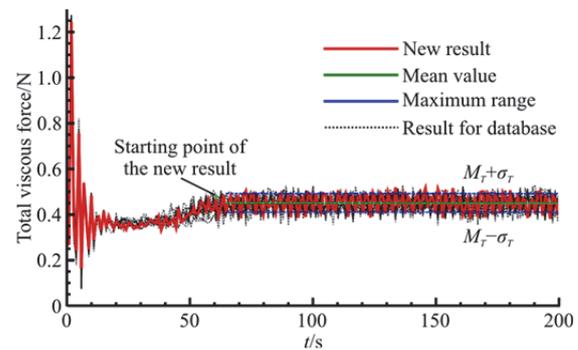


Fig. 13 (Color online) Illustration of the reference database and the tested result for the FSI problem. The gray dashed lines represent multiple results used for generating the database, and the red line is a new result after code modification. The green line shows the time-averaged mean of multiple results, and the blue ones indicate the variation range constrained by the variance

method is used to perform the regression test for this variable. Since the force in the  $y$ -direction is relatively small, only the  $x$ -direction force was considered. Figure 13 displays the reference database and one tested new result for this case. Table 2 shows the converged metric values in the database as well as the ones from the new result. It indicates the new mean is quite close to the converged one, and the new variance is also smaller than the reference one. Therefore, the new result of the FSI problem after code modification is still considered correct.

In general, each test case should have at least one variable of interest used for the regression test, and the testing strategy is not fixed. This regression test environment provides flexible combinations of variables and strategies, but each variable should always have the best option to check itself.

**Table 2 Mean and variance in the reference database and for the new result**

Metric	Database	New result
Mean	0.45570	0.45490
Variance	0.00173	0.00157

## 5. Conclusion

This paper introduces a method for developing an automatic regression test environment for open-source scientific computing libraries, with SPHinXsys as an illustration to demonstrate its functionality. In the context of scientific computing libraries undergoing centralized development, ensuring the accuracy of simulation results is crucial, and the regression test serves as a key procedure in achieving this. The reference database for each benchmark test is generated using different strategies, and the new result after code modifications can be automatically tested against them upon source code updates. This regression test environment has been successfully implemented in all test cases released in SPHinXsys, showcasing its effectiveness in validating new results obtained after code modifications. The initiative aims to raise awareness within the general scientific computing communities regarding the importance of software performance during development. The proposed regression test principle is universal and can be applied and extended to other libraries and applications. Future work will involve the implementation of additional regression test methods, and as the number of test cases grows due to the addition of new dynamics features, the selection and reduction of test cases will be considered.

## Acknowledgments

This work was supported by the China Scholarship Council (Grant No. 202006230071), the Deutsche Forschungsgemeinschaft (DFG) (Grant No. DFG HU1527/12-4). The corresponding code of this

work is available on Git- Hub at <https://github.com/Xiangyu-Hu/SPHinXsys>.

## Compliance with ethical standards

**Conflict of interest:** The authors declare that they have no conflict of interest. All authors declare that there are no other competing interests.

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

**Informed consent:** Not application.

## References

- [1] Drake J. B., Jones P. W., Carr Jr G. R. Overview of the software design of the community climate system model [J]. *The International Journal of High Performance Computing Applications*, 2005, 19(3): 177-186.
- [2] Post D. E., Kendall R. P. Software project management and quality engineering practices for complex, coupled multiphysics, massively parallel computational simulations: Lessons learned from ASCI [J]. *The International Journal of High Performance Computing Applications*, 2004, 18(4): 399-416.
- [3] Oberkampf W. L., Roy C. J. Verification and validation in scientific computing [M]. Cambridge, UK: Cambridge University Press, 2010.
- [4] Farrell P. E., Piggott M. D., Gorman G. J. et al. Automated continuous verification for numerical simulation [J]. *Geoscientific Model Development*, 2011, 4(2): 435-449.
- [5] Kanewala U., Bieman J. M. Testing scientific software: A systematic literature review [J]. *Information and software technology*, 2014, 56(10): 1219-1232.
- [6] Rimmel H., Paech B., Bastian P. et al. System testing a scientific framework using a regression-test environment [J]. *Computing in Science and Engineering*, 2011, 14(2): 38-45.
- [7] Bojana K. Framework for developing scientific applications [D]. Skopje, Macedonia: Cyril and Methodius University, 2018.
- [8] Ural H., Yenigün H. Regression test suite selection using dependence analysis [J]. *Journal of Software: Evolution and Process*, 2013, 25(7): 681-709.
- [9] Agrawal A. P., Choudhary A., Kaur A. An effective regression test case selection using hybrid whale optimization algorithm [J]. *International Journal of Distributed Systems and Technologies (IJDST)*, 2020, 11(1): 53-67.
- [10] Di Nardo D., Alshahwan N., Briand L. et al. Coverage-based regression test case selection, minimization and prioritization: A case study on an industrial system [J]. *Software Testing, Verification and Reliability*, 2015, 25(4): 371-396.
- [11] Prasad S., Jain M., Singh S. Regression optimizer a multi coverage criteria test suite minimization technique [J]. *International Journal of Applied Information Systems*, 2012, 1(8): 5-11.
- [12] Harikarthik S. K., Palanisamy V., Ramanathan P. Optimal test suite selection in regression testing with testcase prioritization using modified Ann and Whale optimization algorithm [J]. *Cluster Computing*, 2019, 22: 11425-11434.
- [13] Tahat L., Korel B., Harman M. et al. Regression test suite prioritization using system models [J]. *Software Testing*,

- Verification and Reliability*, 2012, 22(7): 481-506.
- [14] Lin X., Simon M., Niu N. Exploratory metamorphic testing for scientific software [J]. *Computing in Science and Engineering*, 2018, 22(2): 78-87.
- [15] Lin X., Simon M., Niu N. Hierarchical metamorphic relations for testing scientific software [C]. *Proceedings of the International Workshop on Software Engineering for Science*, Gothenburg, Sweden, 2018.
- [16] Peng Z., Lin X., Niu N. Unit tests of scientific software: A study on SWMM [C]. *Computational Science-ICCS 2020: 20th International Conference*, Amsterdam, The Netherlands, 2020.
- [17] Peng Z., Lin X., Simon M. et al. Unit and regression tests of scientific software: A study on SWMM [J]. *Journal of Computational Science*, 2021, 53: 101347.
- [18] Liu Z., Brinckman K., Dash S. Automated validation of CFD codes for analysis of scramjet propulsive flows using CRAVE [C]. *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Denver, USA, 2009.
- [19] Happ H. J. Linux C-shell regression testing for the SHAMRC CFD code [C]. *Users' Group Conference*, Portland, USA, 2011.
- [20] Zhang C., Rezavand M., Zhu Y. et al. SPHinXsys: An open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics [J]. *Computer Physics Communications*, 2021, 267: 108066.
- [21] Liechtenecker H. P., Riebl R. Towards a modern CMake workflow [C]. *Proceedings of the 8th OMNeT++ Community Summit*, Virtual Summit, 2021.
- [22] Baxter R. Software engineering is software engineering [C]. *26th International Conference on Software Engineering, W36 Workshop Software Engineering for High Performance System (HPCS) Applications*, Edinburgh, UK, 2004.
- [23] Lucy L. B. A numerical approach to the testing of the fission hypothesis [J]. *Astronomical Journal*, 1977, 82: 1013-1024.
- [24] 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.
- [25] 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.
- [26] Zhang C., Rezavand M., Hu X. Dual-criteria time stepping for weakly compressible smoothed particle hydrodynamics [J]. *Journal of Computational Physics*, 2020, 404: 109135.
- [27] Zhang C., Zhu Y., Yu Y. et al. A simple artificial damping method for total Lagrangian smoothed particle hydrodynamics [J]. *Engineering Analysis with Boundary Elements*, 2022, 143: 1-13.
- [28] Zhang C., Rezavand M., Hu X. A multi-resolution SPH method for fluid-structure interactions [J]. *Journal of Computational Physics*, 2021, 429: 110028.
- [29] Zhang C., Wang J., Rezavand M. et al. An integrative smoothed particle hydrodynamics method for modeling cardiac function [J]. *Computer Methods in Applied Mechanics and Engineering*, 2021, 381: 113847.
- [30] Bilotta G., Hérault A., Cappello A. et al. GPUSPH: A smoothed particle hydrodynamics model for the thermal and rheological evolution of lava flows [J]. *Geological Society, London, Special Publications*, 2016, 426(1): 387-408.
- [31] Gomez-Gesteira M., Rogers B. D., Crespo A. J. C. et al. SPHysics—development of a free-surface fluid solver—Part 1: Theory and formulations [J]. *Computers and Geosciences*, 2012, 48: 289-299.
- [32] Crespo A. J. C., Domínguez J. M., Rogers B. D. et al. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH) [J]. *Computer Physics Communications*, 2015, 187: 204-216.
- [33] Cercos-Pita J. L. AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL [J]. *Computer Physics Communications*, 2015, 192: 295-312.
- [34] Springel V. The cosmological simulation code GADGET-2 [J]. *Monthly Notices of the Royal Astronomical Society*, 2005, 364(4): 1105-1134.
- [35] Hopkins P. F. A new public release of the GIZMO code [R]. arXiv preprint arXiv:1712.01294, 2017.
- [36] Zhang C., Hu X., Adams N. A. A generalized transport-velocity formulation for smoothed particle hydrodynamics [J]. *Journal of Computational Physics*, 2017, 337: 216-232.
- [37] Zhang C., Wei Y., Dias F. et al. An efficient fully Lagrangian solver for modeling wave interaction with oscillating wave surge converter [J]. *Ocean Engineering*, 2021, 236: 109540.
- [38] Sakoe H., Chiba S. Dynamic programming algorithm optimization for spoken word recognition [J]. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978, 26(1): 43-49.
- [39] Myers C., Rabiner L., Rosenberg A. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition [J]. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980, 28(6): 623-635.
- [40] Cheng J., Wei F., Liu Y. et al. Chinese sign language recognition based on DTW-distance-mapping features [J]. *Mathematical Problems in Engineering*, 2020, 2020(1): 1-13.
- [41] Li H., Liu J., Yang Z. et al. Adaptively constrained dynamic time warping for time series classification and clustering [J]. *Information Sciences*, 2020, 534: 97-116.
- [42] Douglass A. C. S., Harley J. B. Dynamic time warping temperature compensation for guided wave structural health monitoring [J]. *IEEE transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2018, 65(5): 851-861.
- [43] Al-Naymat G., Chawla S., Taheri J. Sparsedtw: A novel approach to speed up dynamic time warping [C]. *AusDM 2009: Proceedings of the Eighth Australasian Data Mining Conference*, Melbourne, Australia, 2009.
- [44] Ratanamahatana C. A., Keogh E. Everything you know about dynamic time warping is wrong [C]. *Third Work-Shop on Mining Temporal and Sequential Data*, Washington DC, USA, 2004.
- [45] Salvador S., Chan P. Toward accurate dynamic time warping in linear time and space [J]. *Intelligent Data Analysis*, 2007, 11(5): 561-580.
- [46] Keogh E., Ratanamahatana C. A. Exact indexing of dynamic time warping [J]. *Knowledge and Information Systems*, 2005, 7: 358-386.
- [47] Lemire D. Faster retrieval with a two-pass dynamic-time-warping lower bound [J]. *Pattern Recognition*, 2009, 42(9): 2169-2180.
- [48] Sakurai Y., Yoshikawa M., Faloutsos C. FTW: Fast similarity search under the time warping distance [C]. *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Baltimore, USA, 2005.
- [49] Whittaker J. A., Arbon J., Carollo J. How Google tests software [M]. Boston, USA: Addison-Wesley Professional, 2012.