Smoothed Particle Hydrodynamics για ροές με ελεύθερη επιφάνεια

PhD Thesis-Διδακτορική Διατριβή

ΕΥΣΤΑΘΙΟΣ ΧΑΤΖΟΓΛΟΥ

UNIVERSITY OF THESSALY, VOLOS, GREECE
DEPARTMENT OF CIVIL ENGINEERING

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

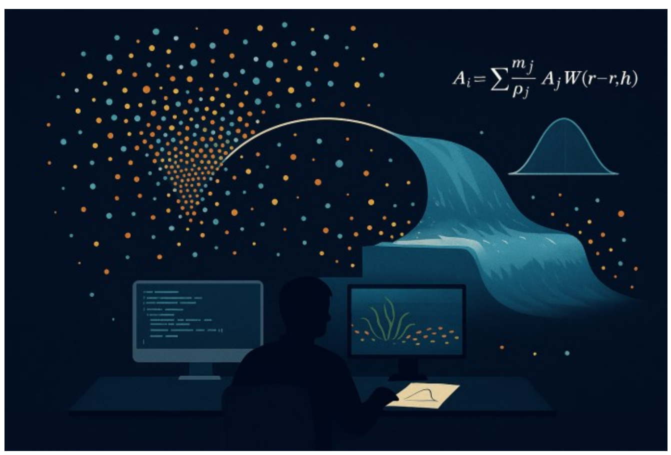
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

PhD Thesis-Διδακτορική Διατριβή

Smoothed Particle Hydrodynamics για ροές με ελεύθερη επιφάνεια

ΕΥΣΤΑΘΙΟΣ ΧΑΤΖΟΓΛΟΥ



Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του

Διδακτορικού Διπλώματος

ΒΟΛΟΣ 2025

“δὶς ἐς τὸν αὐτὸν ποταμὸν οὐκ ἂν ἐμβαίης.”

“You cannot step twice into the same river; neither the river nor you are the same.”

Heraclitus of Ephesus (Cratylus by Plato, 402a)

Enhancing Free Surface Flow Modeling through Smoothed Particle Hydrodynamics-Smoothed Particle Hydrodynamics για ροές με ελεύθερη επιφάνεια

<u>Τριμελής Συμβουλευτική Επιτροπή:</u>

1. Αντώνιος Λιακόπουλος, Ομότιμος Καθηγητής Τμήματος Πολιτικών Μηχανικών, Π.Θ. (Επιβλέπων).
2. Θεόδωρος Καρακασίδης, Καθηγητής Τμήματος Φυσικής, Π.Θ.
3. Ευάγγελος Κεραμάρης, Αναπλ. Καθηγητής Τμήματος Μηχανικών Περιβάλλοντος, ΔΙ.ΠΑ.Ε.

<u>Επταμελής Εξεταστική Επιτροπή:</u>

1) Αντώνιος Λιακόπουλος, Ομότιμος Καθηγητής Τμήματος Πολιτικών Μηχανικών, Π.Θ.
2) Θεόδωρος Καρακασίδης, Καθηγητής Τμήματος Φυσικής, Π.Θ.
3) Ευάγγελος Κεραμάρης, Αναπλ. Καθηγητής Τμήματος Μηχανικών Περιβάλλοντος, ΔΙ.ΠΑ.Ε.
4) Νικήτας Μυλόπουλος, Καθηγητής Τμήματος Πολ. Μηχ., Π.Θ.
5) Χρυσή Λασπίδου, Καθηγήτρια Τμήματος Πολ. Μηχ., Π.Θ.
6) Βασίλειος Κανακούδης, Καθηγητής Τμήματος. Πολ. Μηχ., Α.Π.Θ
7) Φίλιππος Σοφός, Αναπλ. Καθηγητής Τμήματος Φυσικής, Π.Θ.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διδακτορικής διατριβής, θα ήθελα να εκφράσω από καρδιάς τις θερμότερες ευχαριστίες μου προς τον επιβλέποντα μου, ομότιμο καθηγητή Αντώνιο Λιακόπουλο, για την εμπιστοσύνη που έδειξε στο πρόσωπό μου, καθώς και για την αμέριστη υποστήριξή του, τόσο σε ακαδημαϊκό όσο και σε προσωπικό επίπεδο, καθ’ όλη τη διάρκεια εκπόνησης της παρούσας εργασίας. Η καθοδήγησή του υπήρξε καθοριστική όχι μόνο για την επιστημονική πρόοδο του έργου, αλλά και για την προσωπική μου εξέλιξη. Επιπλέον, θα ήθελα να ευχαριστήσω τα υπόλοιπα μέλη της τριμελούς συμβουλευτικής επιτροπής, τον καθηγητή Θεόδωρο Καρακασίδη και τον αναπληρωτή καθηγητή Ευάγγελο Κεραμάρη, για την εξίσου σημαντική βοήθεια και στήριξη αυτά τα έτη. Ιδιαίτερες ευχαριστίες οφείλω και στον αναπληρωτή καθηγητή Φίλιππο Σοφό η βοήθεια του οποίου ήταν καθοριστική για την ολοκλήρωση της παρούσας διατριβής. Θα ήθελα να ευχαριστήσω επίσης όλα τα μέλη της επταμελούς εξεταστικής επιτροπής για την βοήθεια και τον χρόνο τους. Ιδιαίτερη μνεία οφείλω και στους συναδέλφους Δρ. Γεώργιο Σοφιάδη και Απόστολο Παλάση καθώς και τους φίλους μου για την έμπρακτη στήριξή τους κατά τη διάρκεια αυτής της πορείας.

Όλα τα παραπάνω ενδεχομένως δεν θα είχαν καταστεί εφικτά χωρίς τη θεσμική, τεχνική και οικονομική στήριξη του Πανεπιστημίου Θεσσαλίας. Συγκεκριμένα θα ήθελα να εκφράσω την ευγνωμοσύνη μου στο Πανεπιστήμιο και την Επιτροπή Ερευνών για την χορήγηση υποτροφίας στο πλαίσιο «operating framework of the University of Thessaly Innovation, Technology Transfer Unit and Entrepreneurship Center "One Planet Thessaly", under the “Scholarship Grants to University of Thessaly Doctoral Candidates” and was funded by the Special Account of Research Grants of the University of Thessaly.» Την λήξη της οποίας ακολούθησε η ενίσχυση από το ΕΛΙΔΕΚ στο πλαίσιο «Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers” (Project Number: 4584).»

Κλείνοντας, θέλω να εκφράσω την απέραντη αγάπη και ευγνωμοσύνη μου προς την οικογένειά μου. Η στήριξη, η υπομονή και η πίστη τους σε εμένα υπήρξαν οι σταθερές μου σε αυτό το δύσκολο αλλά όμορφο ταξίδι. Χωρίς εκείνους, τίποτα δεν θα ήταν δυνατό.

Ευστάθιος Χατζόγλου

List of Publications

Academic Journals

Chatzoglou, E., & Liakopoulos, A. (2025). Modeling turbulent flow in vegetated open channels using smoothed particle hydrodynamics. Submitted for review.

Chatzoglou, E., Sofiadis, G., & Liakopoulos, A. (2025). Dynamic Smagorinsky model implementation in a LES-SPH approach for turbulent flows. Under review.

Chatzoglou, E., & Liakopoulos, A. (2025). Turbulent flow through sluice gate and weir using smoothed particle hydrodynamics: Evaluation of turbulence models, boundary conditions, and 3D effects. *Water*, *17*(2), 152.

Sofos, F., Sofiadis, G., **Chatzoglou, E**., Palasis, A., Karakasidis, T. E., & Liakopoulos, A. (2024). From sparse to dense representations in open channel flow images with convolutional neural networks. *Inventions*, *9*(2), 27. https://doi.org/10.3390/inventions9020027.

Chatzoglou, E., Liakopoulos, A., & Sofos, F. (2023). Smoothed particle hydrodynamics-based study of 3D confined microflows. *Fluids*, *8*(5), 137. https://doi.org/10.3390/fluids8050137.

Sofos, F., **Chatzoglou, E.**, & Liakopoulos, A. (2022). An assessment of SPH simulations of sudden expansion/contraction 3-D channel flows. *Journal of Computational Particle Mechanics*, *8*(1), 27-45. https://doi.org/10.1007/s40571-021-00396-z

International Conferences

Chatzoglou, E., & Liakopoulos, A. (2025). Smoothed particle hydrodynamics simulation of 3D open channel flow over vegetated bed. Paper presented at the *19th SPHeric World Conference*, Barcelona, Spain.

Chatzoglou, E., & Liakopoulos, A. (2023). Flow regimes in sluice gate-weir system: 3D SPH-based model validations. In *Proceedings of the 17th SPHeric International Workshop*, Rhodes, Greece.

Chatzoglou, E., & Liakopoulos, A. (2022). Simulation of open channel flow control by Smoothed Particle Hydrodynamics. Paper presented at the *Protection and Restoration of the Environment XVI*, Kalamata, Greece.

Chatzoglou, E., & Liakopoulos, A. (2022). Hydraulic jump simulation via Smoothed Particle Hydrodynamics: A critical review. Paper presented at the *39th IAHR World Congress – From Snow to Sea*, Granada, Spain.

Chatzoglou, E., Sofos, F., & Liakopoulos, A. (2021). SPH-based study of confined microflows characterized by abrupt changes in cross-sectional areas. In *Proceedings of the VII International Conference on Particle-Based Methods (PARTICLES 2021)*, Hamburg, Germany.

National Conferences (Greek)

Χατζόγλου, Ε., & Λιακόπουλος, Α. (2022). Η μέθοδος SPH στην υδραυλική ανοιχτών αγωγών: Δύο προβλήματα αναφοράς και αποτίμηση της μεθόδου. Εισήγηση στο *15ο Πανελλήνιο Συνέδριο της Ελληνικής Υδροτεχνικής Ένωσης (ΕΥΕ)*, Θεσσαλονίκη, Ελλάδα.

Contents

List of Acronyms

Acronym	Full Form
BCs	Boundary Conditions
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DBC	Dynamic Boundary Condition
DEM	Discrete Element Method
DNS	Direct Numerical Simulation
DPD	Dissipative Particle Dynamics
FDM	Finite Differences Method
FEM	Finite Element Method
Fr	Froude Number
FSI	Fluid–Structure Interaction
FVM	Finite Volume Method
GPU	Graphics Processing Unit
HPC	High-Performance Computing
k-ε	Turbulence model (k-epsilon)
k-ω	Turbulence model (k-omega)
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
LES-SPH	Large Eddy Simulation integrated with SPH
MD	Molecular Dynamics
mDBC	Modified Dynamic Boundary Condition
NS	Navier–Stokes
OVITO	Open Visualization Tool (used for particle visualization)
RANS	Reynolds-Averaged Navier–Stokes
Re	Reynolds Number
SDPD	Smoothed Dissipative Particle Dynamics
SGS	Sub-grid-Scale
SPH	Smoothed Particle Hydrodynamics
SPHERIC	Smoothed Particle Hydrodynamics European Research Interest Community
SPS	Sub-Particle Scale
VB	ViscoBound (parameter)

List of Figures

Figure 103. Vertical plane at y=0.456 m showing velocity field and streamlines around a vegetation cylinder. (a) x-component velocity distribution along the vertical plane, highlighting the flow, wake formation, and velocity gradients upstream and downstream of the cylinder. (b) Streamlines illustrate flow separation, recirculation, and vortex formation downstream the rigid cylinder and water distortion upstream.

Figure 104. Comparison of velocity profiles extracted from SPH simulations, experiments (Dunn et al., 1996), and finite volume simulations (Kasiteropoulou et al., 2017). SPH results closely match experimental data, demonstrating the method's accuracy for vegetated open channel flows.

Figure 105. Comparison of relative errors (%) between the experimental data (Dunn et al., 1996), results from other conventional numerical methods (Kasiteropoulou et al., 2017) and the SPH results over the range $0 \leq z \leq 0.120$ m. The plot illustrates the deviation of each dataset from the experimental data as a function of height (z), showing how the relative errors vary for each dataset.

Figure 106. Comparison of velocity profiles extracted from SPH simulations with both Standard and Dynamic Smagorinsky models, experiments (Dunn et al., 1996), and finite volume simulations (Kasiteropoulou et al., 2017). SPH-Dynamic Smagorinsky results slightly better match the experimental data, demonstrating the method's accuracy for vegetated open channel flows and the effectiveness of Dynamic Smagorinsky implementation described in section 4.

Figure 107. Velocity profile from z= 0 to 0.12 m (the top of the rigid cylinders).

Figure 108. Comparison of relative errors (%) between the experimental data (Dunn et al., 1996), Kasiteropoulou et al. data, SPH- Standard Smagorinsky simulation results and SPH-Dynamic Smagorinsky simulation results over the range $0 \leq z \leq 0.120$ m.

List of Tables

Smoothed Particle Hydrodynamics για ροές με ελεύθερη επιφάνεια

ΕΥΣΤΑΘΙΟΣ ΧΑΤΖΟΓΛΟΥ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ, ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Περίληψη

Η παρούσα διδακτορική διατριβή παρουσιάζει μια ολοκληρωμένη έρευνα σχετικά με τις δυνατότητες, την εξέλιξη, τους περιορισμούς και τις εφαρμογές της μεθόδου Smoothed Particle Hydrodynamics (SPH). Η μέθοδος SPH αποτελεί ένα εργαλείο προσομοίωσης, στο οποίο το μέσο διακριτοποιείται μέσω ενός συνόλου κινούμενων κόμβων (γνωστών ως σωματίδια). Η Λαγκραζιανή προσέγγιση σε συνδυασμό με την απουσία πλέγματος (grid) προσφέρει σημαντικά πλεονεκτήματα έναντι των παραδοσιακών μεθόδων Υπολογιστικής Ρευστομηχανικής (CFD). Ωστόσο, ως μια μέθοδος που έχει αναπτυχθεί σημαντικά τις τελευταίες δεκαετίες, η SPH εξακολουθεί να αντιμετωπίζει προκλήσεις, δημιουργώντας ευκαιρίες για περαιτέρω βελτίωση. Οι κύριες προκλήσεις περιλαμβάνουν τη μείωση των αριθμητικών σφαλμάτων, τη διαχείριση των συνοριακών συνθηκών, με έμφαση στα σταθερά όρια (solid boundaries), τη μοντελοποίηση τυρβώδους ροής και τη βελτίωση της υπολογιστικής αποδοτικότητας.

Η διατριβή περιλαμβάνει εκτενή θεωρητική και υπολογιστική ανασκόπηση της μεθόδου, εντοπίζοντας τα αδύναμα σημεία της και επιχειρώντας τη βέλτιστη προσομοίωση προβλημάτων με τις υπάρχουσες δυνατότητες της μεθόδου. Για να πραγματοποιηθούν οι παρακάτω προσομοιώσεις επιλέχθηκαν οι κώδικες (open source codes) LAMMPS και DualSPHysics. Η επιλογή αυτή βασίστηκε στη δημοφιλία και την αποδοτικότητά τους. Μέσα από ποικίλα απλά και σύνθετα προβλήματα αποδεικνύεται ότι η μέθοδος SPH είναι ένα αποτελεσματικό εργαλείο προσομοίωσης υδραυλικών προβλημάτων, ιδίως σε ροές ανοιχτών αγωγών και σε προβλήματα με περίπλοκες γεωμετρίες. Τα αποτελέσματα συγκρίνονται με αναλυτικές λύσεις, πειραματικά δεδομένα και αριθμητικές προσομοιώσεις που υπάρχουν στη βιβλιογραφία.

Η κύρια συνεισφορά της διατριβής έγκειται στην ενσωμάτωση ενός μοντέλου υπολογισμού τύρβης (turbulence model) με την ονομασία Dynamic Smagorinsky model. Το μοντέλο αυτό, το οποίο βασίζεται στη μέθοδο Large Eddy Simulation (LES),

ενσωματώθηκε επιτυχώς στον πηγαίο κώδικα DualSPHysics, εξαλείφοντας την ανάγκη εμπειρικής ρύθμισης των παραμέτρων τύρβης και προσφέροντας ένα προσαρμοστικό, δυναμικό μοντέλο. Οι συγκρίσεις με τις προϋπάρχουσες δυνατότητες της μεθόδου σε τυρβώδεις ροές ανέδειξαν τη σημαντική βελτίωση στην ακρίβεια των αποτελεσμάτων.

Εν συνεχεία εξετάστηκε η πρακτική εφαρμογή της μεθόδου σε ένα πρόβλημα τρισδιάστατης ροής ενός ανοιχτού αγωγού με παρουσία βλάστησης στον πυθμένα. Το παραπάνω είναι ένα πρόβλημα με πολύπλοκες συνοριακές συνθήκες και άμεση σημασία για την οικο-υδραυλική και τη μοντελοποίηση περιβαλλοντικών συστημάτων. Τα αποτελέσματα επιβεβαιώνουν την ικανότητα της μεθόδου SPH να αναπαριστά σύνθετες ροές με μεγάλη ακρίβεια. Μάλιστα, μετά την ενσωμάτωση του Dynamic Smagorinsky model, τα αποτελέσματα παρουσιάζουν μικρότερη απόκλιση από τα πειραματικά δεδομένα σε σχέση με παραδοσιακές μεθόδους που βασίζονται σε πλέγμα και τα αποτελέσματα της SPH πριν την ενσωμάτωσή του Dynamic Smagorinsky model.

Συνολικά, η διατριβή συμβάλλει στην προώθηση της SPH ως ένα αξιόπιστο και ευέλικτο εργαλείο Υπολογιστικής Ρευστομηχανικής. Μέσα από συγκριτική αξιολόγηση, βελτιώσεις και πρακτικές εφαρμογές, τοποθετείται το θεμέλιο για μελλοντική έρευνα και βιομηχανική αξιοποίηση. Επιπλέον, αναδεικνύονται δυνατότητες για περαιτέρω καινοτομία, όπως η ενσωμάτωση τεχνητής νοημοσύνης, μηχανικής μάθησης και υβριδικών μοντέλων προσομοίωσης (LES-SPH).

Λέξεις κλειδιά: SPH, Ροή με Ελεύθερη Επιφάνεια, Μοντελοποίηση Τυρβώδους Ροής, Υπολογιστική Ρευστομηχανική (CFD), Dynamic Smagorinsky model, Αριθμητική Προσομοίωση, Υδραυλική Μηχανική, Περιβαλλοντικές ροές.

Enhancing Free Surface Flow Modeling through Smoothed Particle Hydrodynamics

EFSTATHIOS CHATZOGLOU

UNIVERISTY OF THESSALY, CIVIL ENGINEERING DEPARTMENT

Abstract

This thesis presents a comprehensive investigation into the capabilities, limitations, and advancement of the Smoothed Particle Hydrodynamics (SPH) method for simulating fluid flows in hydraulic and environmental engineering contexts. As a mesh-free, Lagrangian approach, SPH offers distinct advantages in handling complex geometries, free surfaces, and multiphase flows, making it a compelling alternative to traditional mesh-based Computational Fluid Dynamics (CFD) techniques. However, the method still faces several challenges, including numerical consistency, boundary condition implementation, turbulence modeling, and computational efficiency.

The research begins with an extensive theoretical and computational review of SPH, identifying its grand challenges and critically assessing existing formulations and kernel functions. Two widely used open-source SPH solvers, LAMMPS and DualSPHysics, are employed to simulate a broad set of test cases, including laminar and turbulent flows in both 2D and 3D configurations, under closed and open channel conditions. Validation is performed against analytical solutions, numerical benchmarks, and experimental data, with results demonstrate that SPH can produce accurate and physically consistent predictions when properly configured. A significant area of focus is the treatment of boundary conditions, a critical factor affecting SPH stability and accuracy. The study evaluates multiple boundary handling strategies, with particular emphasis on the Modified Dynamic Boundary Conditions (mDBC) approach, which provides superior performance in reducing pressure artifacts and enhancing particle behavior near solid interfaces. The effectiveness of this approach is confirmed through geometrically complex flow simulations, including expansions, contractions, and simulations of flow control systems such as sluice gates and weirs.

The key contribution of this thesis is the implementation of a Dynamic Smagorinsky turbulence model into the DualSPHysics source code. This approach eliminates the need

for empirical calibration of turbulence parameters, offering an adaptive, locally responsive model that improves the accuracy of SPH in simulating turbulent regimes. Comparative results show notable improvements in velocity profile predictions, particularly in near-wall regions, and demonstrate closer alignment with theoretical log-law behavior.

The practical applicability of SPH is further demonstrated through simulations of 3D open channel flow over submerged vegetation, a scenario with direct relevance to ecohydraulics and environmental modeling. The SPH results show strong agreement with experimental and finite volume data, affirming the method's capability to resolve complex fluid–structure interactions in natural flow environments. The modified SPH model outperformed both the traditional grid-based numerical method and the existing SPH model.

Overall, this thesis contributes to the advancement of SPH as a robust, flexible, and scalable CFD tool. Through rigorous validation, model development, and practical application, it sets the foundation for future research and industrial deployment of particle-based flow solvers. It also highlights avenues for further innovation, including adaptive resolution methods, machine learning integration, and hybrid numerical frameworks.

Keywords: Smoothed Particle Hydrodynamics (SPH), Free Surface Flow, Turbulence Modeling, Computational Fluid Dynamics (CFD), Dynamic Smagorinsky Model, Boundary Condition Handling, Numerical Simulation, Hydraulic Engineering, Environmental Flows.

1 Introduction

1.1 Motivation

Studying a fluid dynamics problem can be approached through several methods, each offering unique insights and advantages. These methods are generally classified as theoretical, experimental, or computational. Theoretical analysis serves as an essential reference. However, solving complex equations such as the Navier–Stokes (NS) equations is extremely difficult without simplifications. Experimental studies are also a very reliable method to investigate real-life scenarios. Physical models are tested in controlled environments to study various properties (aerodynamic; hydrodynamic etc.). However, for most engineering problems, the required scales make experiments expensive and difficult to build and maintain. In addition, measurements may have errors (deviations from reality) due to the measurement process. Measuring properties in certain regions of the flow (especially in highly turbulent or multiphase flows) can be difficult. There may also be limitations on the spatial and temporal resolution of the measurements, restricting the ability to capture fine-scale phenomena. Despite the limitations, experimental methods remain a critical component of fluid dynamics research and engineering applications. The third approach has gained increasing attention in recent years, mainly due to advances in computational power. Computational methods, particularly in our field of research known as Computational Fluid Dynamics (CFD), are numerical techniques used to solve and analyze problems involving fluid flows.

CFD encompasses a variety of approaches, each with its strengths and limitations. One such approach is the Finite Volume Method (FVM), where the fluid domain is discretized into small control volumes, and the integral form of the governing equations is solved over these volumes (Eymard et al., 2000). The Finite Element Method (FEM) approximates the governing partial differential equations (PDEs) by discretizing the domain into finite elements and solving the weak form of the equations (Huebner et al., 2001). Lattice Boltzmann Method (LBM) is characterized by a kinetic based approach where the fluid is modelled on a lattice grid and the Boltzmann transport equation is solved for particle distribution functions (Krüger et al., 2017). Spectral methods solve the governing equations by expanding the solution in terms of global basis functions (e.q Fourier series, Chebyshev polynomials, Legendre polynomials) (Shen et al., 2011) while Direct Numerical Simulation (DNS) solved the NS equations without any turbulence modeling. It resolved all scales of motions, and it is one of the most accurate CFD methods (Lee and Moser, 2015; Moin and Mahesh, 1998). Large-Eddy-Simulation (LES) resolves large-scale eddies directly while smaller-scale turbulence is modelled using sub-grid scale models (Lesieur et al., 2005; Boris et al., 1992). Reynolds Averaged Navier-Stokes (RANS) is used in many engineering applications due to their lower computational cost. RANS solve the time-averaged NS with turbulent models (k-ε, k-ω etc.) to account for the effect of turbulence (Alfonsi, 2009).

Despite the variety of available CFD techniques, mesh-based approaches often face challenges when dealing with free-surface flows, complex geometries, and highly dynamic interfaces. There are also some particle-based methods such as molecular dynamics (MD) (Plimpton, 1995), dissipative particle dynamics (DPD) (Pivkin et al., 2010), smooth dissipative particle dynamics (SDPD) (Ellero and Español, 2017) and SPH which do not require a grid and have the advantage in certain type of problems. Specifically, SPH is a mesh-free method where the domain is divided into a set of mass-points, usually referred to as particles, and the fluid properties are smoothed over a certain distance. SPH can be used effectively in cases characterized by complex geometries, free-surface flows, multiphase flows and problems with large deformations and complex boundaries (Violeau, 2012).

The ability of SPH to combine other methods (such as LES and DNS), along with its natural handling of free surfaces and complex geometries without the need for a mesh, makes it a very promising computational tool for real-life engineering problems. Nevertheless, SPH still faces some limitations, which, in the author's opinion, can be overcome. Ideally SPH can become a tool that is suitable for the computation of fluid dynamics in different scales, characterized not only by laminar but also highly turbulent flows and without demanding vast computational costs. In addition, the accuracy of the method needs to be tested in different problems to ensure that the SPH method produces accurate qualitative and quantitative results.

This research will present an analytical preview of the SPH method, its most important improvements over the last year and the current grand challenges. It will validate the method by testing different problems of closed and open channels in 2D and 3D and with a great variety of Reynolds numbers. In addition, this research tries to evolve the SPH method by modifying a current code and advancing its capability to simulate turbulent flows. The outcome of this research will be valuable to other researchers who may work with the SPH in early or advanced stages and for diverse applications from microfluidics to hydraulic engineering and environmental flow modeling.

Despite the recent advances, significant gaps remain in accurately simulating complex turbulent flows using the SPH method, particularly for applications involving intricate boundary conditions, adaptive turbulence modeling, and flows in realistic environmental scenarios. Existing implementations often struggle with numerical instabilities, pressure inaccuracies at boundaries, and limited adaptability in turbulent conditions. This thesis aims explicitly to address these limitations by validating and enhancing SPH methods, focusing on improved boundary condition treatments and the integration of adaptive turbulence models within practical engineering contexts.

1.2 SPH grand challenges

While SPH presents significant advantages, its widespread adoption is still limited by several technical and numerical challenges. Identifying and addressing these key challenges in SPH is crucial to effectively extend the method's applicability in complex engineering scenarios, which is a central focus of this thesis. Thus, in this extended review, we list the key grand challenges of SPH as defined by the SPHERIC committee (Vacondio et al., 2021).

- Convergence, Consistency, and Stability

Ensuring numerical consistency and convergence is challenging, especially with irregular particle distributions and kernel-dependent approximations. Stability issues, such as pressure oscillations and particle disorder, persist in weakly compressible formulations (Gotoh et al., 2016).

- Boundary Conditions

Representing boundaries without a structured mesh is difficult, leading to inaccuracies such as spurious forces, pressure leakage, and penetration issues near solid or free-surface boundaries (English et al., 2022; Fourtakas et al., 2019; Wang et al., 2019). In this thesis, particular emphasis is given on improving boundary conditions by evaluating and applying existing techniques.

- Adaptivity & Variable Resolution

Achieving adaptive refinement and variable resolution while maintaining consistency and stability is a major hurdle, especially when efficiently resolving sharp gradients or critical regions (Gao et al., 2022; Yang et al., 2021).

- Coupling to Other Methods

Integrating SPH with other numerical methods, such as Finite Element Methods (FEM), for multi-physics problems like fluid-structure interaction, heat transfer, or phase change remains complex and computationally demanding (Ye et al., 2019; Wang et al., 2016).

- Applicability to Industry

High computational cost, scalability limitations, and difficulties in modeling complex, real-world engineering problems hinder widespread adoption of SPH in industry. Addressing these challenges is crucial to making SPH practical for industrial applications (Shadloo et al., 2016).

Although significant research has advanced SPH methods, the explicit incorporation of adaptive turbulence modeling approaches, such as Dynamic Smagorinsky, remains insufficiently explored. Addressing this gap could substantially improve SPH's practical effectiveness, particularly for engineering applications involving complex turbulent flows.

1.3 Smoothed Particle Hydrodynamics (SPH) method- Literature review

This section reviews significant developments and existing limitations of the SPH method, highlighting critical contributions that form the theoretical and practical basis for the advancements presented in this thesis.

The SPH method was developed in the late 70s by Monaghan for treating the behavior of the systems existing in Astrophysics. This need derived from the limitations of (Eulerian) mesh-based methods due to the lack of defined boundaries (Lucy, 1977; Gingold and Monaghan, 1977). Even though the SPH method meant to be used in Astrophysics as of the early 90s started to be used in hydrodynamics with great prospects. It is a mesh-free method that describes fluid flow by following the motion of fluid particles (point masses). The information from a mathematical model of a moving continuum medium, usually expressed through partial differential equations, is transferred to several neighboring particles within a specified range (Zhang et al., 2018). These particles act as "nodes" for calculating any field variable, based on an approximation that uses a kernel function (Monaghan 1992; Monaghan and Gingold 1983). Various SPH formulations have been developed, ranging from the original Lagrangian formulation to recent Eulerian variants (Nasar et al., 2021). Eulerian methods typically use a fixed reference coordinate system, where fluid properties are transferred between cells. In contrast, Lagrangian methods involve a moving coordinate system in which elements, represented as numerical cells or particles, move along their trajectory while maintaining consistent properties. Mixed Eulerian-Lagrangian approaches combine elements of both Eulerian and Lagrangian frameworks (Shadloo et al., 2016).

Generally, in meshless particles methods the discrete flow is represented by replacing the grid with a finite number of nodes (particles) which carry the fluid characteristic properties. Similarities can be found in other particle methods such as molecular dynamics (MD) for micro-scale applications (Plimpton 1995), dissipative particle dynamics (DPD) (Pivkin et al., 2010) for meso-scale applications, and smooth dissipative particle dynamics (SDPD) (Ellero and Espanol, 2017). Hybrid or multiscale methods also proposed (Albano et al., 2021). However, SPH is a numerical approximation of the partial differential equations of continuum mechanics, while MD, DPD, and SDPD can be applied in flows where continuum theory may or may not be valid (Liakopoulos et al., 2017; Sofos et al., 2013). As mentioned earlier, among the main attractions of SPH is its ability to predict highly strained motions without the use of mesh or grid, based on a collection of mass points (particles) (Violeau, 2012). Each particle carries properties such as mass, position, velocity, and thermodynamic quantities. The core idea is to approximate field variables and their spatial derivatives using a kernel function that smooths particle properties over a finite distance (Gingold and Monaghan, 1977). The impact of the kernel function treatment is crucial for methods consistency given the switching from a continuous formalism to a discrete set for modeling the derivation operators occurring in the equation of motion (Violeau, 2012). Commonly used kernels include the Gaussian, cubic spline, and quintic spline functions (Liu and Liu, 2003; Colagrossi and Landrini, 2003).

The first application of SPH in fluid dynamics can be found back to 1994 (Monaghan, 1994) for the simulation of a free surface problem. As in all numerical methods, SPH produced several challenges during the last decades. Some of them have been improved while others are still developed. One of the first was the concern of stability and more specific pairing instability. This has been overcome up to a certain degree with the introduction of the Wendland kernel (Wendland, 1995). Its impact demonstrated in more recent years by Dehnen and Aly (2012). Another popular approach to improve SPH method's accuracy is the kernel correction technique (Bonet and Lok, 1999). This is attributed to the close relation between the calculation of the smoothing function (the kernel gradient) and the approximation accuracy of the derivatives. More specifically, in their work (Bonet and Lok, 1999) restored the accuracy of a conventional kernel utilizing a corrective kernel gradient. In a more recent study, Huang et al., (2016) proposed a kernel gradient free (KGF) kernel correction with a novel discrete scheme of Laplacian operator for viscous incompressible flows. However, more research on kernels brings to the surface another major issue, tensile instability (Swegle et al., 1995). Tensile instability in SPH occurs under tensile (negative pressure) stress, causing particle clumping and numerical inaccuracies. It arises from SPH kernel gradient approximations amplifying perturbations (Gray et al., 2001). In order to alleviate this issue (Colagrossi, 2005; Monaghan, 2000) introduced a corrective term into the momentum equation. In later studies (Sun et al., 2018) a modification of the SPH operators has been introduced depending on the pressure field value. Influenced by this research (Sun et al., 2019; Lind et al., 2012) particle redistribution techniques have been proposed which in the SPH code may occur as "shifting". These techniques not only help with stability issues but also play a crucial role in the most recent SPH simulations. Another popular approach was the introduction of numerical diffusion. The most well-known is the artificial viscosity scheme proposed by Monaghan (Monaghan and Pongracic, 1985). Another, less popular, is the approach of numerical diffusion through Rieman-SPH introduced by Vila (1999).

With respect to the second grand challenge, convergence, the kernel approximation techniques applied for SPH accuracy are also utilized regarding methods converge. In addition, the particle redistribution techniques have been proved very helpful. Adaptive resolution techniques, including variable smoothing lengths and dynamic particle refinement, have also been employed to allocate higher resolution in regions with steep gradients or shocks, thereby improving convergence in highly dynamic scenarios (Muta & Ramachandran, 2022). Recent advancements have introduced higher-order consistent

SPH formulations, such as the Moving-Least-Squares SPH (MLS-SPH), which improves the accuracy of gradient evaluations and achieves better convergence rates in comparison to traditional SPH methods (Eirís et al., 2023). Similarly, novel divergence, and curl-free formulations have been developed to address inconsistencies in the treatment of fluid vorticity and pressure fields (Frontiere et al., 2017). Hybrid approaches, which combine SPH with grid-based methods or mesh-free finite volume schemes, have also emerged as effective strategies for improving convergence in complex multi-scale problems, such as astrophysical and multiphase flow simulations (Springel, 2010; Hopkins, 2015). Machine learning methods have recently been incorporated to optimize kernel functions and dynamically adjust particle positions, showing promise in reducing numerical artifacts and accelerating convergence (Dong et al., 2024). Despite these advancements, challenges remain, particularly in ensuring general convergence across diverse applications and complex geometries, making it an active and evolving area of research.

Consistency is linked to a correct implementation of BCs, which is also one of the hardest to deal with SPH grand challenges. Several approaches have been proposed in the literature to enforce solid boundaries. One approach is using the method of repulsive forces (Monaghan and Kajitar 2009; Monaghan 1994), yet the accuracy of SPH spatial interpolation close to the solid walls is reduced. Another option is to discretize the boundaries with the so-called semi-analytical formulation (Leroy et al., 2014; Mayrhofer et al., 2013; Kulasegaram et al., 2004). A third way is based on fictitious particles to fill the space beyond the boundary interface to mimic the solid wall (Fourtakas et al., 2019; Adami et al., 2012; Marrone et al., 2011; Crespo et al., 2007). The mDBCs introduced by English et al. (2022), which can alleviate limitations such as dissipation and unphysical values of density and pressure close to the solid boundaries, are based to this third option. The main idea behind mDBCs is that the density of solid particles is obtained from positions within the fluid domain (ghost nodes) by linear extrapolation.

Another challenge regarding the SPH method is the ability to deal with different particle sizes. This is included in the adaptivity grand challenge. The importance of adaptivity comes along with the reduction of the overall computation time while maintaining an elevated level of accuracy. A variant resolution approach has been introduced by Gingold and Monaghan, (1982) for astrophysics problems. Yet their approach cannot be adopted when extending these models in a weakly compressible context since density should remain nearly constant. Recently, several approaches have been developed that dynamically increase or decrease particle size (Vacondio et al., 2013). Chiron et al. (2018) used Adaptive Particle Refinement (APR) methodology based on the use of different layers. SPH methods adaptivity is an ongoing project which produces more encouraging results in the present time.

A different approach to overcome SPH potential limitations while taking advantage of its strengths is to use SPH where it is powerful and then use another method to solve different physics. This raises another SPH potential challenge, how the method couples with other methods. Some of the works that successfully coupled SPH with other methods are Fourey et al. (2017) with the coupling of SPH with FEM, Canelas et al. (2017) coupled SPH with DEM. A numerical method used to model the behavior of granular materials, where the material is represented as a collection of discrete particles. DEM calculates particle movements based on forces like contact, gravity, or external influences. Verbrugghe et al. (2018) coupled SPH with Finite Difference Method (SPH-FDM). Other research includes external coupling in which motion response is influenced by the SPH solution (Bulian and Cercos-Pita, 2018; Serván-Camas et al., 2016). Xu et al. (2021) proposed a coupling between SPH and Finite Volume Methods, which is a very promising approach since FVM is advantageous in problems dealing with complex fluid interfaces. The integration of Artificial Intelligence (AI) and Machine Learning (ML) into SPH has enabled significant advancements in simulation accuracy and efficiency. Machine learning techniques, such as graph neural networks (GNNs), have been used to model fluid dynamics, enhancing predictions in complex systems (Toshev et al., 2024). Active learning methods have been applied to refine particle distributions dynamically, improving resolution in critical regions (Dong et al., 2024). Additionally, ML-based optimization of kernel functions and reinforcement learning for adaptive particle refinement have reduced numerical artifacts and computational costs (Ladický et al., 2015). These integrations demonstrate the potential of AI to make SPH simulations more robust and scalable across various fields.

All the above challenges are encountered for the ultimate purpose of a CFD tool and thus SPH. It is an important pulse to bring SPH into the industrial context. Therefore, a grand challenge is the applicability of the method to actual, real-life engineering problems. SPH has been successfully applied to several fluid dynamics problems in cases of industrial use such as coastal engineering (Gomez-Gesteira et al., 2012). Several studies were successfully conducted with the SPH method regarding water wave impacts on offshore and coastal structures or even floating objects (Altomare et al., 2015; Lind et al., 2015; Altomare et al., 2014; Ni et al., 2014). Other studies present applications on river engineering and environmental flows such us river waterworks, spillways, flows under sluice gates and over weirs and fish passes (Chatzoglou and Liakopoulos, 2025; Chatzoglou and Liakopoulos, 2023; English et al., 2022; Novak et al., 2021; Tafuni et al., 2016, Husain et al., 2014) and Sloshing of liquids (Zhang et al., 2024; Marrone et al., 2023; Shao et al., 2012). SPH has also been applied to problems of multiphase flows such as bubble dynamics, sediment transport and liquid-gas interactions (Feng et al., 2024; Zhang et al., 2024; Zhang et al., 2015; Manenti et al., 2012; Hu and Adams, 2006). All the above studies and many others focus on how a researcher, or an engineer can use SPH not only for academic-research purposes but also for an engineering tool.

At an early stage of this work, we observed that SPH lacks accurate prediction of turbulent flows in many cases. Modeling turbulence in SPH is a challenging topic due to the method's inherited limitations. Some of the basic limitations are the lack of consistent variable resolution, the order of accuracy of SPH spatial operators and complex interactions between fluid and solid particles (Ricci et al., 2023). One of the earliest attempts to simulate turbulence with SPH was proposed by Monaghan (2002). In Monaghan's work an adaptation of the Lagrangian-averaged Navier Stokes (LANS) α-model has been proposed for SPH. Lo and Shao, (2002) simulated the near-shore solitary wave mechanics combining LES with SPH, achieving good results regarding wave profiles. Their model extended by (Dalrymple & Rogers, 2006) where the Standard Smagorinsky model implemented into the DualSPHysics source code. They validated their results against numerical experiments of wave overtopping, beach waves and dam breaking flow, achieving good accuracy. Violeau and Issa, (2007) studied turbulent free surface flows with the SPH method implementing k-ε and Standard Smagorinsky models. Their results showed a good correlation with the log law. Later studies such as Mayrhofer et al. (2015) coupled LES with SPH for studying the turbulent channel flow yielded in overprediction of the of the streamwise velocity. Kazemi et al. (2020) simulated turbulent flows with and over natural gravel beds achieving reliable results. In all the above studies it is proven that SPH formulation is characterized by a kernel function analogous to the filter function of length (grid filter in LES) and thus SPH-LES simulations can easily be related. Choi and Kim, (2023) simulated lid-driven flow under transient and turbulent conditions using LES-SPH approach. Other studies such as (Ricci et al., 2023; Mayrhofer et al., 2015; Robinson and Monaghan, 2012) combined SPH with DNS achieving very good results. It is evident that the scientific community of SPH is increasingly focusing on blending SPH with methods like LES and/or DNS to achieve better qualitative and quantitative results when highly turbulent flows are studied. Hence, it is obvious that alternative models or alternative implementation in different code environments will improve the way that SPH method handles turbulence, leading to more accurate results, even in complex flows.

Although previous studies have significantly advanced SPH, important gaps remain, particularly regarding accurate turbulence modeling and boundary condition implementation. This thesis directly addresses these gaps through the implementation and rigorous validation of the Dynamic Smagorinsky turbulence model within the DualSPHysics environment, as well as through detailed evaluations and enhancements of boundary conditions, thus significantly extending the method's applicability and accuracy.

In the context of this study the applicability of SPH in actual problems relevant to hydraulics and environmental engineering is crucial. Thus, on the one hand we focus on the deep understanding of the SPH method and its validation with experimental data and theoretical predictions and on the other, we focus on the development of the method in as many grand challenges as possible in the context of this research. A major part of our research effort is dedicated to the modification of the open-source code

DualSPHysics. By the end of this study, we will reach into the conclusion that SPH is a reliable simulation tool yet still needs to face some of the limitations mentioned earlier. To put our brick in this method's growth we proved how accurate treatment of the existing solid boundary conditions treatment can produce optimal results and we implemented a turbulence model into the DualSPHysics code.

1.4 Thesis Objectives

Considering the limitations and gaps identified in previous sections, particularly related to BCs accuracy, turbulence modeling, and applicability to complex hydraulic scenarios, this thesis aims to advance the SPH method by addressing these specific challenges through targeted computational experiments, code modifications, and rigorous validation.

The main objectives of this work are to evaluate SPH simulations in the context of fluid dynamics and to improve the effectiveness of the SPH method in simulating turbulent complex flows. Several cases have been tested for closed and open channel flows, from a variety of problems. Simulations on different scales, with various boundary conditions and flow characteristics, will be tested and compared with analytical solutions, numerical results and experimental data from the literature.

The objectives of this thesis are:

- Perform rigorous validation of SPH using benchmark analytical solutions, numerical methods, and experimental data to assess and confirm its accuracy in hydraulic and environmental applications.
- Evaluate, enhance, and validate boundary condition treatments in SPH, particularly focusing on mDBCs, to mitigate known limitations such as numerical instabilities and pressure inaccuracies.
- Implement and rigorously validate the Dynamic Smagorinsky turbulence model within the DualSPHysics environment to provide accurate, adaptive, and calibration-free turbulence modeling capabilities in SPH.
- Demonstrate the practical applicability and robustness of the enhanced SPH methods by successfully simulating complex, realistic hydraulic engineering problems, specifically flows under sluicegates and over weirs and open channel flows with vegetated beds.

Achieving these objectives will help establish SPH as a reliable, robust, and widely applicable CFD tool for hydraulic and environmental engineering.

1.5 Thesis Layout

The thesis is structured as follows:

Chapter 2 covers the detailed SPH formulation and the mathematical background of the method. A detailed literature review is given on previous and recent SPH applications in open and closed channel flows, with emphasis on turbulent flows and implementation milestones for the method. Finally, this chapter presents the two main SPH codes used for numerical experiments in this study (LAMMPS and DualSPHysics).

In Chapter 3, several test cases are presented to evaluate the performance of the SPH method. Extensive work on solid boundary conditions and turbulence modeling with the current versions of the SPH codes revealed certain limitations. One of the most significant challenges identified is turbulence modeling, which led the author to implement a code modification, detailed in Chapter 4.

Chapter 4 introduces the implementation of an LES approach, the Dynamic Smagorinsky model, into DualSPHysics code. It includes a detailed explanation of the Dynamic Smagorinsky, its SPH formulation and how it can be implemented in a SPH solver. The performance of the implementation is examined through test cases in which a previous turbulence model and Dynamic Smagorinsky are compared.

After validating the method and implementing the Dynamic Smagorinsky turbulence model, Chapter 5 presents a more complex case: a turbulent open channel flow with submerged vegetation. This case is particularly relevant to eco-hydraulics and environmental flow studies.

Chapters 6 and 7 conclude the findings of the study and propose ideas for future work.

Overall, this thesis aims to validate and enhance the SPH method's performance for hydraulic and environmental applications, with focus on boundary condition treatment, turbulence modeling, and practical applicability to real-world engineering problems.

2 Principles of the SPH method

This chapter presents a detailed overview of the fundamental principles and mathematical framework of the SPH method. Key elements such as kernel functions, boundary condition handling, and temporal integration schemes are outlined, emphasizing aspects directly relevant to the improvements introduced in this thesis.

The SPH method uses a two-step interpolation process (Violeau, 2012). First, a continuous function is approximated using an integral formulation, where the function A(r, t) is convolved with the Dirac delta function δ over the domain Ω

$$A(r,t) = (A * \delta)(r,t) = \int_{\Omega} A(r',t)\delta(r - r')d\Omega \tag{2-1}$$

where r' is another node (or particle) in the domain. Then the Dirac delta function is replaced by a smooth kernel function W(r−r', h), which introduces a smoothing length h. This leads to the smoothed approximation. The basic idea of the SPH method is to approximate the δ-Dirac function through a smoothing kernel W.

2.1 Kernel Function

The effectiveness of the SPH numerical method is significantly influenced by the selection of the smoothing kernel. These kernels are defined as functions of the non-dimensional distance between particles (q), calculated as $q = r/h$. Here, r represents the distance between any two particles a and b, while the parameter h, known as the "smoothing length," determines the extent of the area around particle a within which neighboring particles are considered (Violeau, 2012). The smoothing kernel must satisfy several key properties, as outlined by Monaghan (1992) and Liu (2003), such us positivity within a defined interaction zone, compact support, normalization, a monotonically decreasing value with distance, and differentiability. For a more detailed explanation of SPH, readers can refer to the work of Monaghan (2005). A typical kernel graphical representation can be seen in Figure 1.

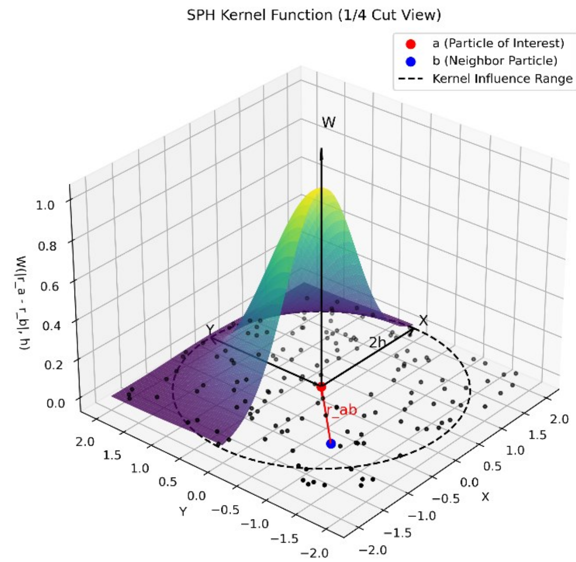


Figure 1. Typical kernel function. The region where a particle affects the interpolated value.

The interpolating kernel (W) should satisfy two properties (Violeau, 2012)

$$\int W(r - r', h)\, dr' = 1 \tag{2-2}$$

known as zeroth moment, and

$$\int (r - r')\, W(r - r', h)\, dr' = 0 \tag{2-3}$$

called the first zeroth moment.

Using a discrete approximation, the integral in Eq. 2-1 can split up into Rieman summations approximating the function as

$$A_s(r) = \sum_b m_b \frac{A_b}{\rho_b} W(r - r_b, h) \tag{2-4}$$

where m_b is the mass of particle b while ρ represents the density.

Selecting an appropriate kernel is very important for the accuracy and efficiency of the SPH method. A number of kernel functions have been proposed in the literature. We list below the most used.

2.1.1 Gaussian Kernel

One of the earliest kernels in the SPH method is the Gaussian Kernel. The original calculations of Gingold and Monaghan (1997) used a Gaussian kernel in the form

$$W(r_{ij},\mathrm{h}) = \frac{1}{h^d} f\left(\frac{|r_{ij}|}{h}\right) \tag{2-5}$$

$$f(s) = \frac{1}{h^2\pi} e^{-s^2} \tag{2-6}$$

The Gaussian kernel has compact support over an infinite radius, meaning the computational domain around a particle can theoretically be extended without increasing the smoothing length. However, in practice, the influence of a particle beyond a distance of 4h from the center is negligible. Since the Gaussian kernel is derived from an exponential function, it asymptotically approaches zero outside the domain but never fully reaches it. The Gaussian kernel and its first derivative are depicted in Figure 2.

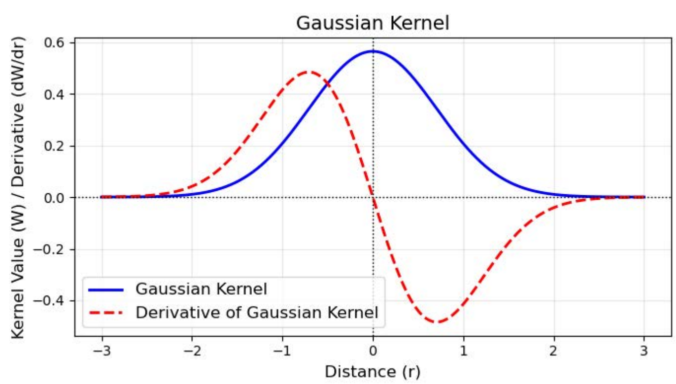


Figure 2. An example of the Gaussian kernel function and its first derivative.

More accurate kernels in one dimension can be constructed by requiring them to be normalized (Monaghan, 1992)

$$\int r^2 W(r,\mathrm{h})\mathrm{dr} = 0 \tag{2-7}$$

The Super Gaussian kernel in three dimensions (Monaghan, 1992) is expressed as

$$W(r,\mathrm{h}) = \frac{1}{\pi^{3/2}h^3}\left(\frac{5}{2} - r^2\right)\left(-\frac{r^2}{h^2}\right) \tag{2-8}$$

$$f(s) = \frac{1}{14\pi}\begin{cases}(2-s)^3 - 4(1-s)^3, & 0 \le s < 1;\\ (2-s)^3, & 1 \le s < 2;\\ 0, & 0 \le s \ge 2\end{cases} \tag{2-9}$$

$$f(s) = \frac{7}{478\pi}\begin{cases}(3-s)^5 - 6(2-s)^5 + 15(1-s)^5, & 0 \le s < 1;\\ (3-s)^5 - 6(2-s)^5, & 1 \le s < 2;\\ (3-s)^5, & 2 \le s < 3;\\ 0 & s \ge 3.\end{cases} \tag{2-10}$$

2.1.2 Cubic Spline

A class of kernels that meet the specified criteria is the polynomial splines. These are piecewise polynomial functions designed to be positive, decreasing, and smoothly defined within the domain while remaining zero elsewhere (English, 2020, thesis). This includes the Cubic spline which is given by

$$W(r,h) = \frac{\sigma}{h^v}\begin{cases}1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if } 0 \le \frac{r}{h} \le 1;\\ \frac{1}{4}(2-q)^2 & \text{if } 1 \le \frac{r}{h} \le 2;\\ 0 & \text{otherwise}\end{cases} \tag{2-11}$$

In Eq. 11, $q=r/h$, where r represents the particle position, v refers to the dimension of the problem while σ equals to $10/7\pi h^2$ in 2D and $1/\pi h^3$ in 3D. The kernel and its first derivative are depicted in Figure 3.

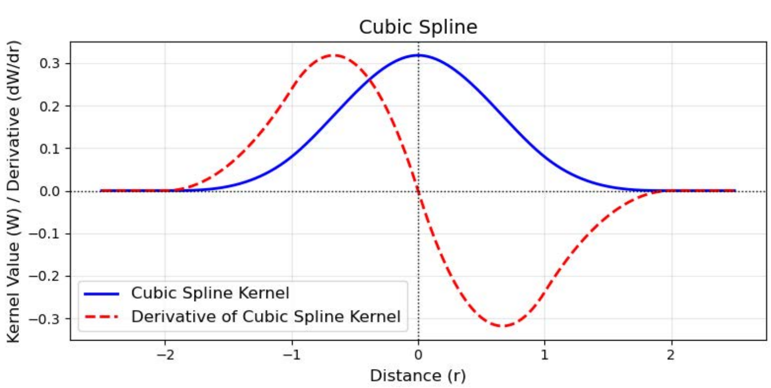


Figure 3. An example of the Cubic spline kernel function and its first derivative.

2.1.3 Wendland kernel

Wendland, (1995) developed a family of polynomial splines which are used as kernels. The most popular among these is the quintic spline, often referred to as Wendland kernel. The Wendland kernel is defined by the function:

$$W(\boldsymbol{r},h) = a_D \left(1 - \frac{q}{2}\right)^4 (2q+1)\ if\ 0 \le q \le 2 \tag{2-12}$$

In Eq. 2-12 where α_D is equal to $7/4\pi h^2$ in 2-D and $21/16\pi h^3$ in 3-D. Wendland's plot and its first derivative are presented in Figure 4.

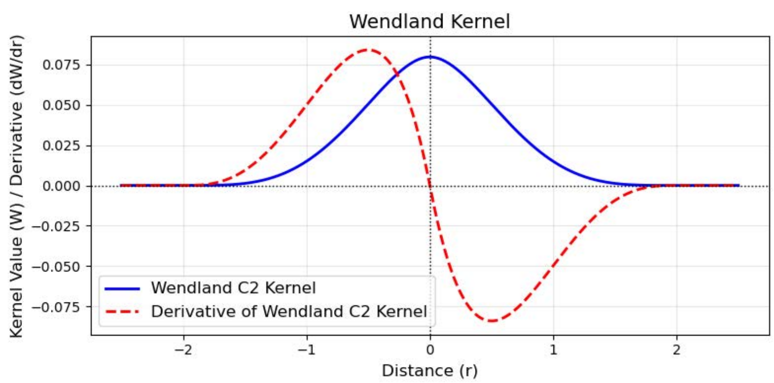


Figure 4. An example of the Wendland kernel function and its first derivative.

2.1.4 Other kernel functions

Mueller et al. (2003) designed kernels for special purposes.

$$W_{\text{poly6}}(r,\text{h}) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \le r \le h \\ 0 & \text{otherwise} \end{cases} \tag{2-13}$$

In Eq. 2-13 the distance r only appears squared meaning that it can be evaluated without computing square roots in distance computations.

$$W_{\text{spiky}}(r,\text{h}) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & 0 \le r \le h \\ 0 & \text{otherwise} \end{cases} \tag{2-14}$$

In Eq 2-14, a kernel introduced by Desbrun (Desbrun and Cani, 1996) is presented. This kernel generated the necessary repulsive forces. Also, it has zero first and second derivatives at the boundaries.

In real-time applications with a relatively low number of particles, this effect can lead to stability issues. To address this, Mueller et al. (2003) developed a third kernel specifically for calculating viscosity forces

$$W_{\text{viscosity}}(r,\mathrm{h}) = \frac{15}{2\pi h^3}\begin{cases} -\frac{r^3}{2\mathrm{h}^3} + \frac{r^2}{h^2} + \frac{h}{2\mathrm{r}} - 1 & 0 \le r \le h \\ 0 & \text{otherwise} \end{cases} \tag{2-15}$$

In Eq. 2-15 the Laplacian of the kernel is positive everywhere with the following additional properties:

$$\nabla^2 W(r,\mathrm{h}) = \frac{45}{\pi h^6}(h - r)$$

$$W(|r| = \mathrm{h},\mathrm{h}) = 0 \tag{2-16}$$

$$\nabla W(|r| = \mathrm{h},\mathrm{h}) = 0$$

Dehnen and Aly (2012) gave a good overview of various functional forms and quantities for the B-Spline equation and Wendland's functions in 1-3 spatial dimensions. To mention a few:

$$\mathrm{WC2}(s) = \frac{7}{\pi}\begin{cases} (1-q)^4(4q+1), & 0 \le q < 1; \\ 0 & q \ge 1 \end{cases} \tag{2-17}$$

$$\mathrm{WC4}(s) = \frac{9}{\pi}\begin{cases} (1-q)^6\left(1 + 6q + \frac{35}{3}q^2\right), & 0 \le q < 1; \\ 0, & q \ge 1 \end{cases} \tag{2-18}$$

$$\mathrm{WC6}(s) = \frac{78}{7\pi}\begin{cases} (1-q)^8(1 + 8q + 25q^2 + 32q^3), & 0 \le q < 1; \\ 0, & q \ge 1 \end{cases} \tag{2-19}$$

2.2 Kernel Corrections

SPH often suffers from inconsistency, meaning that standard formulations cannot exactly reproduce even simple fields (like constant or linear functions) due to particle discretization. Over the years, various kernel correction methods have been developed to restore consistency and improve accuracy

Randles and Libersky (1996) introduced a renormalization approach using a correction tensor (often denoted by B) to improve SPH consistency. It is essentially the inverse of a local moment matrix of particle kernel gradients, constructed to ensure that SPH approximations can reproduce linear field gradients exactly. In mathematical form, the correction tensor for particle i can be defined as:

$$B_i = \left(\sum_j \frac{m_j}{\rho_j} (x_j - x_i) \otimes \nabla W ij \right)^{-1} \tag{2-20}$$

Bonet and Lok (1999) developed a kernel gradient correction technique within a variational SPH framework to achieve first-order completeness in gradient estimates. Their method introduces a correction tensor L that scales the kernel gradient, ensuring that the gradient of any linear field is computed exactly (i.e., enforcing first-order consistency for derivatives). In practice, L_i is defined as the inverse of the first moment matrix of the kernel around particle i:

$$L_i = \left(\sum_j \frac{m_j}{\rho_j} \nabla W ij \otimes (x_j - x_i) \right)^{-1} \tag{2-21}$$

Applying this tensor to the standard SPH gradient formula yields a corrected gradient at particle

$$\nabla f_i = L_i \sum_j \frac{m_j}{\rho_j} \mathrm{f_j} \nabla W ij \tag{2-22}$$

This kernel gradient correction significantly improves accuracy and stability for linear field variations (eliminating errors that occur in uncorrected SPH when particles are irregularly distributed).

Liu and Liu’s (2006) consistency restoration method, known as the Corrective Smoothed Particle Method (CSPM), corrects the kernel interpolation to enforce exact reproduction of constant and linear fields (zeroth- and first-order consistency). This is achieved by incorporating first-order terms from a Taylor series expansion into the SPH approximation while neglecting higher-order terms. The resulting formulation introduces a normalization that resembles Shepard interpolation, ensuring the summation of kernel weights equals unity. For example, a corrected zeroth-order consistent interpolation for a scalar field f is given by:

$$f_i = \frac{\sum_j \frac{m_j}{\rho_j} f_j\, Wij}{\sum_j \frac{m_j}{\rho_j} Wij} \tag{2-23}$$

which guarantees that a constant field is exactly represented (the denominator acts as a normalization factor). CSPM also provides a corrected gradient estimator of similar form, improving first-order accuracy in derivatives.

Sibilla (2015) proposed a correction algorithm to attain second-order consistency in SPH, meaning the scheme can exactly reproduce quadratic variations in the field. This method extends the Taylor-series-based correction approach by including second order (quadratic) terms in the particle approximation, allowing exact interpolation of polynomial fields up to degree two. In effect, the scheme achieves C2 consistency for first derivatives (second-order accurate gradient estimates) and C1 consistency for second derivatives. For example, in one dimension the second-order consistent SPH formula for a first derivative can be written as:

$$\left.\frac{\partial f}{\partial x}\right|_{x_i} = \frac{\sum_j (f_i - f_j)((x_j - x_i)\, Wij}{\frac{1}{2}\sum_j (x_j - x_i)^2\, Wij} \tag{2-24}$$

Here the denominator contains the second-order moment of the kernel weight distribution, which adjusts the gradient estimate to account for curvature in f. By enforcing such second-order moment conditions in the SPH discretization, Sibilla's method yields a spatial convergence of order two for smooth fields, significantly improving accuracy for problems with strong gradients or nonlinear field variations.

2.3 Boundary Conditions

Boundary conditions do not naturally emerge within the SPH formalism. As a fluid particle approaches a solid boundary, only particles within the system contribute to the summations, with no interaction from outside the boundary (Violeau, 2012). This can lead to unrealistic effects because different variables behave differently near boundaries, for instance velocity tends to drop to zero, while density does not. To address boundary issues, various methods involve creating virtual particles to represent the system's boundaries. Three distinct types of solid particles are typically used for this purpose (Crespo et al., 2007). As mentioned before, boundary conditions treatment is still one of the SPH method's grand challenges (Vacondio et al., 2021) and a lot of research groups try to develop new practices or evolve existing ones.

2.3.1 Ghost Particles

The concept of ghost particles in SPH was introduced to address challenges associated with enforcing boundary conditions in simulations. This technique involves placing fictitious particles, mirror images of real particles, beyond the physical boundaries of the simulation domain. These ghost particles help maintain accuracy and stability near boundaries by ensuring consistent kernel support, thereby mitigating kernel truncation issues and enforcing no-slip wall conditions (Randles and Libersky, 1996; Libersky 1991). When a real particle approaches a boundary within a distance shorter than the kernel smoothing length, a virtual (ghost) particle is created outside the system, mirroring the real particle. These particles share the same density and pressure but have opposite velocities. As a result, the number of boundary particles changes with each time step, making the implementation more complex in the code.

$$\begin{aligned}
v_{\text{ghost}} &= 2v|wall - v_{fluid} \\
v(r_w) &= \sum_j \frac{m_j}{\rho_j} v_j W(r - r_j) \\
&= \sum_{j\in\text{Fluid}} \frac{m_j}{\rho_j} v_j W(r - r_j) + \sum_{j\in\text{Ghost}} \frac{m_k}{\rho_k} v_k W(r - r_k) \\
&= \sum_{j\in\text{Fluid}} \frac{m_j}{\rho_j} v_j W(r - r_j) + \sum_{j\in\text{Ghost}} \frac{m_k}{\rho_k} (2v|\text{wall} - \text{v}_{\text{fluid}}) W(r - r_k) \\
&= \sum_{j\in\text{Fluid}} \frac{m_j}{\rho_j} v_j W(r - r_j) + \sum_{j\in\text{Fluid}} \frac{m_j}{\rho_j} (2v|\text{wall} - \text{v}_j) W(r - r_j) \\
&= 2v_{\text{wal}} \sum_{j\in\text{Fluid}} \frac{m_j}{\rho_j} W(r - r_j).
\end{aligned} \tag{2-25}$$

2.3.2 Repulsive Particles

Monaghan (1994) introduced this type of boundary particles. In this approach, the boundary particles exert central forces on the fluid particles, similar to the forces between molecules. For a boundary particle and a fluid particle separated by a distance r, the force per unit mass follows the Lennard-Jones potential.

$$f(r) = \begin{cases} \frac{D}{r}\left(\left(\frac{r_o}{r}\right)^{12} - \left(\frac{r_o}{r}\right)^{6}\right), & 0 < r < r_o \\ 0, & r > r_o \end{cases} \tag{2-26}$$

Similarly, other authors, such as Peskin (1977), describe this force by assuming the presence of forces at the boundaries, which can be represented using a delta function. Monaghan and Kos (1999); Monaghan et al. (2003) later refined this method through an interpolation process, reducing the impact of boundary particle spacing on the wall's repulsive force.

$$\chi(x)\begin{cases}1-\dfrac{x}{p_{\text{sep}}}, & 0\le x<p_{\text{sep}}\\ 0, & \text{otherwise}\end{cases} \tag{2-27}$$

$$\begin{aligned} f = f_{ak_1}(x1,y)+f_{ak_2}(x2y) &= \frac{U_{max}^2}{y\dfrac{2m_b}{m_a+m_b}(y)\big(\chi(x_1)+\chi(x_2)\big)} \\ &= \frac{U_{max}^2}{y\dfrac{2m_b}{m_a+m_b}(y)\left(1-\dfrac{x_1}{p_{\text{sep}}}+1-\dfrac{x_2}{p_{\text{sep}}}\right)} \\ &= \frac{U_{max}^2}{y\dfrac{2m_b}{m_a+m_b}(y)} \end{aligned} \tag{2-28}$$

In Eqs (2-27,2-28), p_{sep} represents the interparticle distance. In many cases it is also called as “dp.” The Function Γ(y) is defined in terms of the variable s=y/h.

$$\Gamma(y)=\begin{cases}\frac{2}{3}, & 0<s<2/3;\\ \frac{5}{14\pi}[3(2-s)^2-12(1-s)^2], & 2/3<s<1;\\ \frac{5}{14\pi}[3(2-s)^2], & 1<s<2;\\ 0, & \text{otherwise}\end{cases} \tag{2-29}$$

While Monaghan’s updated force scheme (Monaghan & Kajtar, 2009)

$$f_{ak}=-f_{ka} \tag{2-30}$$

$$f_{ak}=\frac{K}{\beta}\frac{2m_k}{m_a+m_k}\frac{r_{ak}}{|r_{ak}|^2}W(r_{ak},h) \tag{2-31}$$

In Eq. 2-31, K is an SPH constant (Monaghan et al., 2003).

$$f_x=\sum_{j=-\infty}^{\infty}\frac{x_a-x_j}{r_{aj}}\Phi(r_{aj}) \tag{2-32}$$

$$f_y = \sum_{j=-\infty}^{\infty} \frac{y_a}{r_{aj}} \Phi\left(r_{aj}\right) \quad (2\text{-}33)$$

$$f_x \approx \int_{-\infty}^{\infty} \frac{x_a - q}{r} W(r) dq \quad (2\text{-}34)$$

$$f_y \approx y_a \int_{-\infty}^{\infty} \frac{W(r')}{r'} dq' \quad (2\text{-}35)$$

2.3.3 Dynamic Particles

These particles follow the same continuity and state equations as fluid particles, but their positions remain fixed or are externally controlled. A notable advantage is their computational efficiency, as they can be processed within the same loops as fluid particles, significantly reducing computation time. First introduced by Dalrymple and Knio (2001), these particles have been used in studies on wave interactions with coastal structures. More information about the BCs in each of the two selected codes are described in the following subsections.

2.3.4 Boundary conditions in LAMMPS code

In the SPH method within the LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) package, boundary conditions are crucial for defining the behavior of particles at the edges of the simulation domain. SPH is a meshless, particle-based method, and the implementation of boundary conditions in LAMMPS follows certain conventions to ensure correct simulation results.

In macroscopic simulations, it is often necessary to apply boundary conditions that restrict fluid to a specific area. These boundary conditions need to remain stationary. One approach to creating solid boundaries is by using the various *fix wall/** commands available in the main LAMMPS distribution. However, this method tends to result in poor energy conservation. A more effective alternative is to use stationary SPH particles as boundary conditions, where only the internal energy *E* and local density *rhop* are integrated (Ganzenmüller et al., 2011). The appropriate integration can be achieved with the

fix fix_ID group_ID meso/stationary command.

2.3.4.1 Solid boundaries

Solid boundaries in SPH simulations are often represented by fixed or dynamically updating ghost particles that do not move but interact with fluid particles. These ghost particles can be generated using the fix wall/sph command in LAMMPS, which defines solid boundaries that interact with SPH particles using a repulsive force or other specified interactions. Wall boundaries can also be formed by stationary SPH particles which participate in the SPH summations with zero velocities. However, sums for internal energy per particle and local density values are calculated for wall particles and contribute to total simulation values (Sofos et al., 2022). An example of LAMMPS code with SPH formulation of boundary conditions is:

fix wall1 all wall/sph zlo pos 0.0 density 1000.0

The user sets a solid boundary at the z_{lo} position (lower part of Z axis), with a specific density for the boundary particles. The repulsive interaction is handled internally by LAMMPS based on the equations mentioned above. *wall1* represents the group of boundary particles.

2.3.4.2 Boundary conditions regarding fluid particles

Another special handling is required for boundaries that are not formed by solid particles but the boundaries of the fluid particles, for example, the boundaries at the streamwise direction of an open channel flow with infinite length. There are two main options in LAMMPS. The use of Periodic boundary conditions (PBCs), which means that each particle crosses the domain enters the flow with the exact same properties from the opposite direction or the fixed command where the particle exits the domain is just deleted. The way that this is denoted in the LAMMPS script file is

boundary f p p

Where f stands for fixed and p for periodic. More specifically, this line stands for fixed boundary conditions in x-axis and periodic boundary conditions in y, z axis, respectively.

There are also some other options mentioned by Thompson et al. (2022) which can be found in Table 1.

Table 1. Boundary conditions treatment in LAMMPS code.

p	Periodic
f	non-periodic and fixed
s	non-periodic and shrink-wrapped
m	non-periodic and shrink-wrapped with a minimum value

For style s, the position of the face is adjusted to enclose the atoms within that dimension (shrink-wrapping), regardless of how far they move. However, if there is a substantial difference between the current box dimensions and the shrink-wrapped box dimensions, this can cause atoms to be lost at the start of a parallel run. For style m, shrink-wrapping is applied, but it is constrained by the value specified in the data or restart file or set by the *create_box* command. For instance, if the upper z face has a value of 10.0 in the data file, the face will always be positioned at 10.0 or higher, even if the maximum z-extent of all the atoms falls below 10.0. This approach is useful when starting a simulation with an empty box or when you want to leave space on one side of the box, such as for atoms to evaporate from a surface (Thompson et al., 2022).

2.3.5 Boundary conditions in DualSPHysics code

In DualSPHysics, boundaries are represented by a distinct set of particles, separate from the fluid particles. The software supports the creation of solid impermeable boundaries as well as periodic open boundaries. Additionally, it offers methods to move boundary particles based on predetermined forcing functions.

2.3.5.1 Solid Boundary Conditions

Dynamic Boundary Conditions (DBCs)

The default method for the solid boundary conditions in the DualSPHysics code is the dynamic boundary conditions (DBC) (Crespo et al., 2007). In this method, boundary particles are governed by the same equations as fluid particles, but they do not move in response to the forces acting on them. Instead, they either stay fixed in place or move according to a predefined motion function, such as in the case of moving objects like gates, wave-makers, or floating structures (Dominguez et al., 2022). When a fluid particle approaches a boundary and the distance between the boundary particles and the fluid particle drops below twice the smoothing length (h), the density of the boundary particles increases, leading to a rise in pressure. This pressure increase generates a repulsive force on the fluid particle due to the pressure term in the momentum equation.

The stability of this method depends on the time step being sufficiently short to accommodate the highest velocity of any fluid particles interacting with the boundary particles. This is a crucial factor when determining how the variable time step is calculated.

Modified Dynamic Boundary Conditions (mDBCs)

Due to some limitations of the DBCs, such as over dissipation, evolution of density and pressure of the fixed boundary particles (Dominguez et al., 2022), their use may result in unphysical values close to the walls especially in problems with complex geometries. In addition, in most cases, high resolution fields are needed, and this would lead to large computational times. The no-slip approximation, which involves setting zero velocity at each boundary particle, does not accurately represent the condition at low resolutions. This results in higher-than-expected velocities near the boundary and within the flow. Nonetheless, DBC remains effective for coastal engineering applications because it allows for the discretization of complex 3D geometries without requiring intricate mirroring techniques or semi-analytical wall boundary conditions.

To account for this issued English et al. (2022) proposed the mDBCs. In this implementation, boundary particles are positioned similarly to those in the original DBC, with the boundary interface located half a particle spacing away from the innermost layer of boundary particles. For each boundary particle, a ghost node is projected into the fluid across the boundary interface, following a method akin to Marrone et al. (2011). On a flat surface, the ghost node is mirrored across the boundary interface along the direction of the boundary normal that points into the fluid, and the fluid properties at this ghost node are determined using a corrected SPH sum of the surrounding fluid particles. In the case of a boundary particle at a corner, using the boundary normal is ineffective due to the presence of multiple normals. Instead, the boundary interfaces of each solid boundary converge to form a corner, and the ghost node is mirrored through this corner point into the fluid region. Similarly, fluid properties at the ghost node are obtained via a corrected SPH sum of the surrounding fluid. In Figure 5, the way the normal vector is created and the direction they point are presented.

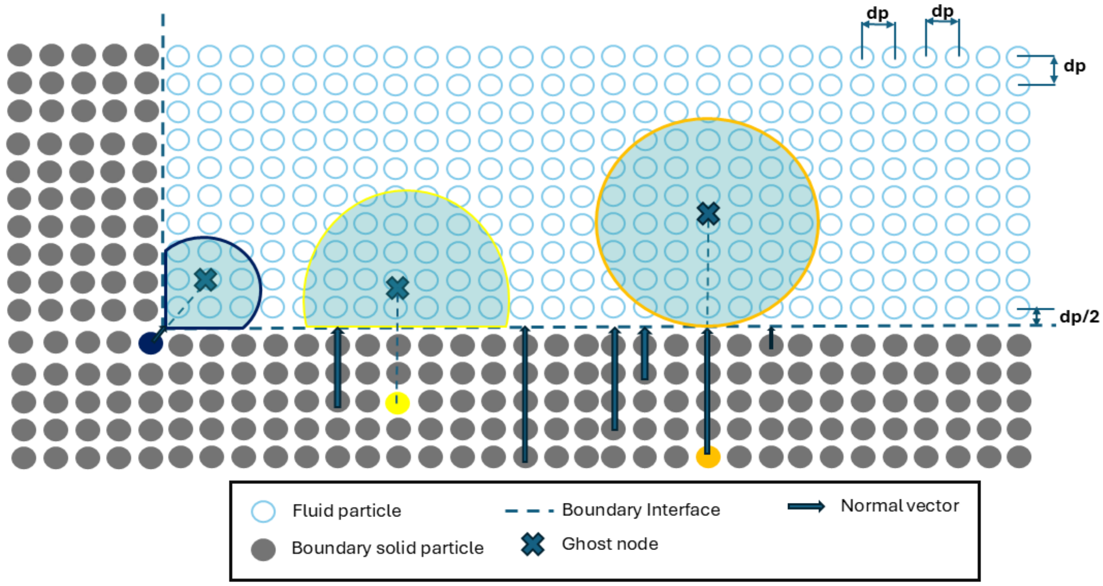


Figure 5. Projection of ghost nodes when the mDBC method is applied.

The boundary particles receive fluid properties using the values calculated at the ghost node and an extrapolation method like the one used for open boundaries in (Tafuni et al. 2018). The density of the boundary particle ρ_b and the ghost particle ρ_g and its gradients are computed at the ghost node using the first order SPH interpolation proposed by (Liu and Liu, 2006). This requires solving the following linear system for each ghost node:

$$\boldsymbol{A}_g \cdot \begin{bmatrix} \rho_g \\ \partial_x \rho_g \\ \partial_y \rho_g \\ \partial_z \rho_g \end{bmatrix} = \boldsymbol{b}_g$$

where:

$$
\boldsymbol{A}_g = \begin{bmatrix} \sum_j W_{gj}V_j & \sum_j (x_j - x_g)W_{ij}V_j & \sum_j (y_j - y_g)W_{gj}V_j & \sum_j (z_j - z_g)W_{gj}V_j \\ \sum_j \partial_x W_{gj}V_j & \sum_j (x_j - x_g)\partial_x W_{ij}V_j & \sum_j (y_j - y_g)\partial_x W_{gj}V_j & \sum_j (z_j - z_g)\partial_x W_{gj}V_j \\ \sum_j \partial_y W_{gj}V_j & \sum_j (x_j - x_g)\partial_y W_{ij}V_j & \sum_j (y_j - y_g)\partial_y W_{gj}V_j & \sum_j (z_j - z_g)\partial_y W_{gj}V_j \\ \sum_j \partial_z W_{gj}V_j & \sum_j (x_j - x_g)\partial_z W_{ij}V_j & \sum_j (y_j - y_g)\partial_z W_{gj}V_j & \sum_j (z_j - z_g)\partial_z W_{gj}V_j \end{bmatrix}
$$

$$
\boldsymbol{b}_g{}^T = [\sum_j W_{gj}m_j \;\; \sum_j \partial_x W_{gj}m_j \;\; \sum_j \partial_y W_{gj}m_j \;\; \sum_j \partial_z W_{gj}m_j
$$

The volume $V_j = \frac{m_j}{\rho_j}$

To evaluate the matrix $\boldsymbol{A}_g$ and the vector $\boldsymbol{b}_g$ at each ghost node g, an additional particle interaction loop must be computed. This new interaction involves inverting the matrix $\boldsymbol{A}_g$, but this process demands only minimal extra computational effort compared to the original DBC formulation. It is also important to note that the matrix $\boldsymbol{A}_g$ is diagonally dominant, meaning it seldom becomes ill-conditioned, even in cases of highly disordered particle distributions (English et al., 2019).

The velocity at the ghost node is found using a Shepard filter sum

$$
\boldsymbol{u}_g = \frac{\sum_j \boldsymbol{u}_j W_{gj} V_j}{\sum_j W_{gj} V_j} \tag{2-36}
$$

When the mDBCs option is applied, several layers of boundary particles must be created. In addition, the normal vectors must be verified because it has been observed that all boundary particles without normal data behave as DBC.

2.3.5.2 Boundary conditions regarding fluid particles

Regarding the treatment of BCs solely for fluid particles, there are two main options withing the DualSPHysics code. The first one is the incorporation of PBCs, which enable fluid particles leaving one side of the simulation domain to reappear on the opposite side. This option is useful for simulating continuous flow or infinite domains (e.g., wave tanks or channels). They are implemented by mapping particle positions from one side to the opposite when they exit the domain. The second option is the open boundaries (or the so-called Inlet/Outlet BCs), which allow the fluid to enter or exit the simulation domain and control velocity, pressure and water depth through buffer zones at the inlet and outlet accordingly. It can be extremely helpful in simulations of a channel with variable cross-section and many other problems which the user requires to imply a constant flow rate. The use of these open boundaries gives a huge advantage on DualSPHysics code.

2.4 Fluid Dynamic equation approximations

In the weakly compressible SPH formulation, Lagrangian form, the Partial Differential Equations (PDEs) of continuity equation and momentum equation are written as

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{v} \tag{2-37}$$

$$\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho}\nabla P + \boldsymbol{g} + \boldsymbol{\Gamma} \tag{2-38}$$

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v} \tag{2-39}$$

where in Eqs. (2-37,2-38 and 2-39), P is the pressure, $\boldsymbol{x}$ is the position vector, $\boldsymbol{g}$ represents the external forces and, **Γ** stands for the dissipative terms in the momentum equation (Monaghan, 1992)

2.4.1 SPH discretization

In SPH the fluid is divided into a set of discrete moving particles. The method interpolates the set of field variables by means of a kernel smoothing function, W. For a function *f*(***r***), we write:

$$f(\boldsymbol{r_i}) = \sum_j m_j \frac{f_j}{\rho_j} W\left(\boldsymbol{r}_i - \boldsymbol{r}_j\right) \tag{2-40}$$

The sum in Eq. 2-40 theoretically extends over all the fluid particles, yet only particles for which $r_i - r_j < h$ need to be considered for appropriately selected kernel function of compact support. This saves computational cost but can lead to inaccuracies close to the boundaries (especially at the wall/fluid interface). m_j, ρ_j represent mass and density of the neighbouring particle at position r_j.

A family of SPH derivative operators can be defined as follows (Ricci et al., 2023)

$$\nabla A = \sum_j V_j \frac{{\rho_j}^{2k} A_i + {\rho_i}^{2k} A_j}{(\rho_i \rho_j)^k} \nabla W_{ij} \tag{2-41}$$

$$\nabla \cdot \boldsymbol{A} = \sum_j V_j \frac{{\rho_j}^{2k}\boldsymbol{A}_i + {\rho_i}^{2k}\boldsymbol{A}_j}{(\rho_i \rho_j)^k} \nabla W_{ij} \tag{2-42}$$

In Eqs (2-41,2-42) the exponent k is chosen either as 0 or 1. Applying these operators to the Eqs (2-37,2-38 and 2-39) we obtain the weakly compressible formulation of the governing equations in SPH weakly compressible formulation.

2.4.2 Weakly compressible SPH approximation equations

Introducing the SPH approximations, the governing equations can be expressed in SPH formation as follows

$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j} \boldsymbol{v}_{ij} \cdot \nabla_i W_{ij} \tag{2-43}$$

$$\frac{d\boldsymbol{v}_i}{dt} = -\sum_j m_j \left(\frac{P_j + P_i}{\rho_j \rho_i}\right) \nabla_i W_{ij} + \boldsymbol{g} + \sum_j m_j \left(\frac{4\nu_0 \boldsymbol{r}_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)({r_{ij}}^2 + \eta^2)}\right) v_{ij} + \boldsymbol{\Gamma}_2 \tag{2-44}$$

where ν_0 is the kinematic viscosity, $\boldsymbol{\Gamma}_2$ represents the dissipation term due to the Sub-Particle Scale (SPS) turbulence model (Dominguez et al., 2022).

2.4.3 Computation of the pressure field

The pressure field is computed through the Tait’s equation of state.

$$P = B\left[\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right] \tag{2-45}$$

In Eq. 2-45, $B = \frac{{c_s}^2 \rho_o}{\gamma}$, ρ_o is the reference density, c_s is the speed of sound and γ=7. The speed of sound should be chosen carefully (normally ten times the maximum speed of the system) in order to limit the density variations to maximum 1% of the reference density (Dominguez et al., 2022).

2.4.4 Density diffusion term

The equation of state represents a highly rigid density field, and due to the inherent disorganization of the particles, the density scalar field exhibits high-frequency, low-amplitude oscillations. As a result, in many SPH simulations density fluctuations that jeopardize simulations’ accuracy have been observed. In order to reduce these fluctuations a density diffusion term has been proposed (Molteni and Colagrossi, 2009). With the diffusive term the continuity equation takes the form

$$\frac{d\rho_i}{dt} = \sum_j \frac{m_j}{\rho_j} \boldsymbol{v}_{ij} \cdot \nabla_i W_{ij} + \delta_\Phi h\, c_0 \sum_j \Psi_{ij} \cdot \nabla_i W_{ij}\, \frac{m_j}{\rho_j} \tag{2-46}$$

with,

$$\Psi_{ij} = 2(\rho_i - \rho_j) \frac{x_{ij}}{\left\| x_{ij} \right\|^2}$$

This formulation, introduced by Molteni and Colagrossi (2009), incorporates a free parameter that requires appropriate assignment. The modification involves adding the Laplacian of the density field to the continuity equation. Antuono et al. (2012) conducted an in-depth analysis of this term's impact on the system by decomposing the Laplacian operator, examining operator convergence, and performing linear stability analysis to assess the influence of the diffusive coefficient.

Within the domain bulk, this equation precisely describes a diffusive term, but its behavior shifts near open boundaries, such as free surfaces. Due to kernel truncation—where no particles exist beyond an open boundary—first-order contributions are nonzero (Antuono et al., 2010), leading to a net force on the particles. However, in nonhydrostatic cases, this effect is negligible, as the resulting force is several orders of magnitude smaller than other dominant forces.

Antuono et al. (2010) proposed corrections for this effect, but they require solving a renormalization problem for the density gradient, which significantly increases computational cost. For most applications, a δ_Φ coefficient of 0.1 is recommended.

In a more recent study Fourtakas et al. (2020) introduced a correction to the previous equation. In their research they substituted the dynamic density with a total one. Hence, the Ψ term takes the form

$$\Psi_{ij} = 2({\rho_i}^D - {\rho_j}^D) \frac{x_{ij}}{\left\| x_{ij} \right\|^2} \tag{2-47}$$

where D denotes the dynamic density of the pressure. Since $\rho_D = \rho_T - \rho_H$, where ρ_T is the total density and ρ_H is the hydrostatic component, the previous equation can take the form

$$\Psi_{ij} = 2({\rho_{ij}}^T - {\rho_{ij}}^H) \frac{x_{ij}}{\left\| x_{ij} \right\|^2} \tag{2-48}$$

In the framework of weakly compressible SPH, the equation of state establishes a relationship between the density and the total pressure at a particle's location. However, only the hydrostatic component of the pressure is required. By applying the equation of state (EOS), the hydrostatic density variation can be determined as follows:

$$\rho_{ij}{}^{H} = \rho_0\left(\sqrt[\gamma]{\frac{P_{ij}{}^{H} + 1}{C_B}} - 1\right) \quad (2\text{-}49)$$

$$P_{ij}{}^{H} = \rho_0 g z_{ij} \quad (2\text{-}50)$$

where z_{ij} is the vertical distance between particle i and particle j and

$$C_B = \frac{c_0{}^2 \rho_o}{\gamma}$$

In the absence of gravity force the formulation collapses to the original formulation of (Molteni and Colagrossi, 2009).

2.4.5 Shifting algorithm

It is common in particle methods, as SPH, that particles may not be able to maintain a uniform distribution, especially in violent flows. As a result, noise occurs in the velocity and pressure field and unexpected voids are created within the fluid flow. To alleviate the anisotropic particle spacing, Xu et al. (2009) proposed a particle shifting algorithm. The shifting algorithm relocates ("shifts") particles toward regions with lower concentration, ensuring a more uniform distribution across the domain and preventing voids. Lind et al. (2012) proposed an improvement on the initial shifting algorithm using Fick's law of diffusion to control the shifting direction and magnitude.

Fick's first law relates the diffusion flux to the concentration gradient, scaled by the diffusion coefficient

$$\boldsymbol{J} = -D_F \nabla C \quad (2\text{-}51)$$

where J is the flux, C is the particle concentration and D_F is the Fickian diffusion coefficient. Using particle concentration, particle shifting distance $\delta \boldsymbol{r}_s$ can be computed by

$$\delta \boldsymbol{r}_s = -D \nabla C_i \quad (2\text{-}52)$$

where D is a new diffusion coefficient that controls the shifting magnitude and absorbs the constant of proportionality. The gradient oof the particle concentration is given by

$$\nabla C_i = \sum_j \frac{m_j}{\rho_j} \nabla_i W_{ij} \tag{2-53}$$

The proportionality of coefficient D is computed through the Skillen et al. (2013) form

$$D \leq \frac{1}{2} \frac{h^2}{\Delta t_{max}} = Ah\|u\|_i dt \tag{2-54}$$

where Δt_{max} is the maximum local time step the is permitted by the CFL condition. A is a dimensionless constant that remains unaffected by the problem setup and discretization, while dt represents the current time step. A ranges from from 1 to 6, with 2 being the default choice.

The shifting algorithm relies heavily on full kernel support. However, particles located at or near the free surface lack complete kernel support, leading to errors in free-surface predictions and potentially causing non-physical instabilities. Directly applying Fick's law would result in rapid diffusion of fluid particles from the bulk due to the steep concentration gradients at the free surface.

To mitigate this issue, Lind et al. (2012) introduced a free-surface correction that restricts diffusion along the surface normal while allowing shifting along the tangential direction. This correction is applied only near the free surface, which is identified based on particle divergence, computed using the following equation originally proposed by Lee et al. (2008).

$$\nabla \cdot \boldsymbol{r} = \sum_j \frac{m_j}{\rho_j} \nabla_i \, \boldsymbol{r}_{ij} W_{ij} \tag{2-55}$$

This idea applied by Dominguez et al. (2022) to Eq. 2-52 multiplying the equation with a free surface correction coefficient (A_{FSC})

$$A_{FSC} = (\nabla \cdot \boldsymbol{r} - A_{FST})/(A_{FSM} - A_{FST}) \tag{2-56}$$

where A_{FST} is the free-surface threshold (1.5 for 2D and 2.75 for 3D), and A_{FSM} is the maximum value of the particle divergence (2 for 2D and 3 for 3D simulations).

2.4.6 SPH treatment of viscosity

The DualSPHysics code provides two viscosity treatment models. "Artificial viscosity (Monaghan, 1992)" and "Laminar+SPS (Dalrymble and Rogers, 2006)."

The artificial viscosity scheme (Monaghan, 1992) is mostly used in SPH simulations due to its simplicity. In SPH formation it is written as

$$\frac{d\boldsymbol{v}_i}{dt} - \sum_j m_j \left(\frac{P_j + P_i}{\rho_j \rho_i} + \Pi_{ij}\right) \nabla_i W_{ij} + \boldsymbol{g} \tag{2-57}$$

where the term Π_{ij} in eq. 15 is given by

$$\Pi_{ij} = \begin{cases} -\dfrac{\alpha \bar{c}_{ij} \mu_{ij}}{\rho_{ij}} & \boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij} < 0 \\ 0 & \boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij} > 0 \end{cases} \tag{2-58}$$

In Eq. 2-58, $\bar{c}_{ij}$ is the mean speed of sound, $\mu_{ij} = \frac{h \boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij}}{r_{ij}^2 + \eta^2}$ and α is a coefficient that needs to be tuned in order to introduce the proper dissipation. A value of α=0.01 has proven to give reliable results in many cases of free surface flows (Altmore et al., 2015), $\eta^2 = 0.01h^2$.

Using the Laminar+SPS (Sub-Particle-Scale) viscosity treatment the momentum equation can be expressed as (Dominguez et al., 2022)

$$\begin{aligned} \frac{d\boldsymbol{v}_i}{dt} \\ = -\sum_j m_j \left(\frac{P_j + P_i}{\rho_j \rho_i}\right) \nabla_i W_{ij} + \boldsymbol{g} \\ + \sum_j m_j \left(\frac{4 \nu_o r_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(r_{ij}^2 + \eta^2)}\right) \boldsymbol{v}_{ij} \\ + \sum_j m_j \left(\frac{\vec{\tau}_{\alpha\beta}{}^j}{\rho_j{}^2} + \frac{\vec{\tau}_{\alpha\beta}{}^i}{\rho_i{}^2}\right) \nabla_i W_{ij} \end{aligned} \tag{2-59}$$

where α=1,2,3, β=1,2,3 and $\vec{\boldsymbol{\tau}}$ is the Sub-Particle Scale stress tensor which is calculated similar to the Standard Smagorinsky method (Sub-grid scale tensor). Favre-averaging is essential for addressing compressibility in weakly compressible SPH as described by Dalrymple and Rogers (2006). In their approach, the eddy viscosity assumption is employed to model the Sub-Particle Scale (SPS) stress tensor. Using Einstein notation, the shear stress component in the coordinate directions i and j is represented:

$$\frac{\vec{\tau}_{ij}}{\rho} = \nu_\tau \left(2S_{ij} - \frac{2}{3} k \delta_{ij}\right) - \frac{2}{2} C_I \Delta^2 \delta_{ij} |S_{ij}|^2 \tag{2-60}$$

In Eq. (2-60) ν_τ is the turbulent eddy viscosity and, k the turbulent kinetic energy. Δ is the initial interparticle distance. C_I=0.0066 is Blin's constant (Blin et al., 2003) and S_{ij} is the Favre-filtered strain rate tensor.

$$S_{ij} = -\frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{2-61}$$

The standard Smagorinsky model (Smagorinsky, 1963) is used for the computation of the eddy viscosity Eq. 2-62.

$$\nu_\tau = (C_s\Delta)^2|S| \tag{2-62}$$

The local strain rate is given as

$$|S| = \sqrt{2S_{ij}S_{ij}} \tag{2-63}$$

where in Eq. 2-63 the Smagorinsky constant C_s=0.12.

2.5 Temporal integration

In order to solve the SPH discrete governing equations various time integration schemes have been proposed. The most popular among them are two explicit time integration schemes. The first is a computational simple Verlet based scheme and the second is the computationally intensive but more stable two-stage Symplectic method (Domingez et al., 2022). In the weakly compressible formulation of SPH the governing partial differential equations (PDEs) are transformed into a set of ordinary differential equations (ODEs). For a representative particle α, the ODEs are written symbolically in the form:

$$\frac{d\rho_a}{dt} = R_a; \ \frac{d\boldsymbol{V}_a}{dt} = \boldsymbol{F}_a; \ \frac{d\boldsymbol{r}_a}{dt} = \boldsymbol{V}_a; \tag{2-64}$$

where an ODE is added for the particle position, $\boldsymbol{r}_a$.

2.5.1 Verlet scheme

The Verlet scheme (Verlet, 1967) is widely utilized in MD due to its low computational cost and second-order spatial accuracy. It also avoids the need for multiple computation steps within each iteration. In this method, the WCSPH variables are calculated according to

$$
\begin{gathered}
v_a^{n+1} = v_a^{n-1} + 2\Delta t\, F_\alpha{}^n \\
r_a^{n+1} = r_a^n + 2\Delta t\, v_\alpha{}^n + \frac{1}{2}\Delta t^2 F_\alpha{}^n \\
\rho_a^{n+1} = \rho_a^{n-1} + 2\Delta t\, R_\alpha{}^n
\end{gathered}
\tag{2-65}
$$

The use of staggered time integration can cause the density and velocity equations to decouple, potentially leading to divergence in the results. To mitigate this, an intermediate correction step is recommended every N_S steps (typically $N_S \approx 40$), as suggested by

$$
\begin{gathered}
v_a^{n+1} = v_a^n + \Delta t\, F_\alpha{}^n \\
r_a^{n+1} = r_a^n + 2\Delta t\, v_\alpha{}^n + \frac{1}{2}\Delta t^2 F_\alpha{}^n \\
\rho_a^{n+1} = \rho_a^n + \Delta t\, R_\alpha{}^n
\end{gathered}
\tag{2-66}
$$

where the subscript *n* denotes the time step and $t = n\Delta t$.

2.5.2 Symplectic scheme

The symplectic position Verlet time integration method (Leimkuhler and Matthews, 2016) achieves second-order accuracy in time. It is particularly well-suited for Lagrangian approaches due to its time-reversibility and symmetry when diffusive terms are absent, preserving the system's geometric properties. Without dissipative forces, the position Verlet scheme is expressed as follows:

$$
\begin{gathered}
r_a^{n+\frac{1}{2}} = r_a^n + \frac{\Delta t}{2} v_\alpha{}^n \\
v_a^{n+1} = v_a^n + \Delta t\, F_\alpha{}^{n+\frac{1}{2}} \\
r_a^{n+1} = r_a^{n+\frac{1}{2}} + \frac{\Delta t}{2} v_\alpha{}^{n+1}
\end{gathered}
\tag{2-67}
$$

In the presence of viscous forces $\mathbf{F}_\alpha$ and density evolution, the velocity is required in n+1/2 step. Thus, a velocity half step is used for the computation of acceleration and density evolutions respectively. The scheme reads

$$
\begin{gathered}
r_a^{n+\frac{1}{2}} = r_a^n + \frac{\Delta t}{2} v_\alpha{}^n \\
v_a^{n+\frac{1}{2}} = v_a^n + \frac{\Delta t}{2} F_\alpha{}^n \\
v_a^{n+1} = v_a^n + \Delta t\, F_\alpha{}^{n+\frac{1}{2}} \\
r_a^{n+1} = r_a^n + \frac{\Delta t}{2} (v_\alpha{}^{n+1} + v_a^n)
\end{gathered}
\tag{2-68}
$$

The density evolution according to Parshikov et al. (2000) follows the half time steps of the symplectic position Verlet scheme as

$$\rho_a^{n+\frac{1}{2}} = \rho_a^n + \frac{\Delta t}{2} R_\alpha{}^n$$
$$\rho_a^{n+1} = \rho_a^n + \frac{2 - \varepsilon\text{\^{}}(\nu + \left(\frac{1}{2}\right) \Delta t}{2} R_\alpha{}^n \qquad (2\text{-}69)$$

Equations. 2-69 are integrated in time by the Symplectic position or Verlet algorithm. The timestep, Δt, is limited by the Courant-Friedrichs-Lewy (CFL) conditions because of the underlying PDEs. Furthermore, the time step depends on the forcing term, $\mathbf{F}_\alpha$, and the viscous diffusion term. Specifically, the timestep satisfies the relations:

$$\Delta t = CFL \min\left(\Delta t_f, \Delta t_{cv}\right) \qquad (2\text{-}70)$$

where

$$\Delta t_f = \min_\alpha \left(\sqrt{\frac{h}{|\boldsymbol{f}_a|}}\right), \text{ and} \qquad (2\text{-}71)$$

$$\Delta t_{cv} = \min_\alpha \left(\frac{h}{c_s + \max_b \left|\frac{h \boldsymbol{v}_a \cdot \boldsymbol{r}_a}{r_{ab}{}^2 + \eta^2}\right|}\right) \qquad (2\text{-}72)$$

and c_s denotes the speed of sound.

2.6 Tensile Instability in SPH

Despite the many advantages of the SPH method, it is susceptible to numerical instabilities under specific stress conditions. One of the most prominent is tensile instability, which occurs when particles are subjected to tensile (negative pressure) stresses (Figure 6). Under such conditions, the standard SPH formulations can cause unphysical particle clustering due to spurious attractive forces, leading to degraded accuracy and stability (Swegle et al., 1995; Monaghan, 2000). Understanding the origins of tensile instability and the methods developed to mitigate it is essential for ensuring the robustness and fidelity of SPH simulations, particularly in problems involving tension-dominated flows or solid mechanics.

The origins of tensile instability are primarily associated with the characteristics of the SPH interpolation kernel and the discretization of the pressure gradient. Negative stresses exacerbate small perturbations in particle distribution, leading to instability growth (Swegle et al., 1995). To mitigate these effects, multiple strategies have been proposed. One influential approach is the artificial stress method introduced by Monaghan (2000), which adds a repulsive correction proportional to the tensile stress. Later improvements, such as the use of Wendland kernels (Dehnen and Aly, 2012), have demonstrated enhanced stability due to their higher-order continuity and positive-definite smoothing properties.

Recent advancements specifically address tensile instability in SPH for both fluids and solids. For instance, Sun et al. (2017) proposed a stress-point based SPH formulation that naturally suppresses tensile artifacts by redefining the interpolation of stresses and forces. Further refinement by Sun et al. (2018) introduced a density-invariant stress correction, significantly improving accuracy in large deformation simulations.

Other advanced methods, such as the Godunov SPH (GSPH) (Murante et al., 2011) and Moving Least Squares SPH (MLS-SPH) (Eirís et., 2023; Tamai and Koshizuka, 2014), inherently address tensile instability by improving the consistency of interpolation and force calculation. Nevertheless, tensile instability remains an active research topic, especially in high-fidelity simulations of fracture, multiphase flows, and solid mechanics.

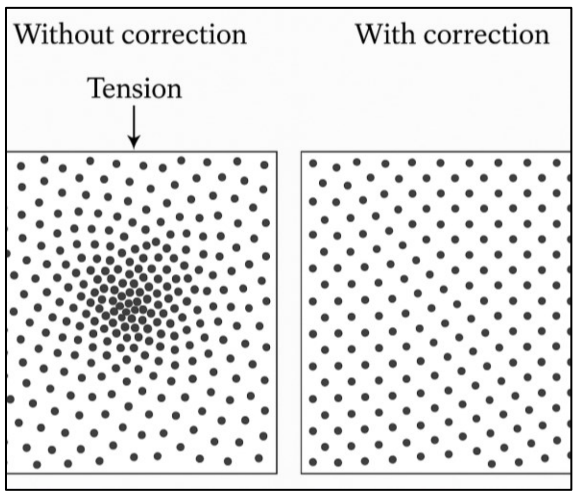


Figure 6. Illustration of tensile instability in SPH particle distributions. Left: Without correction, tensile stresses lead to unphysical particle clustering due to spurious attractive forces. Right: With correction (e.g., artificial stress, improved kernels or shifting).

2.7 SPH code environments

A wide range of code environments has been developed to support SPH simulations across disciplines such as astrophysics, fluid mechanics, and multi-physics engineering. These environments differ in programming language, parallelization strategy, and domain specialization. In astrophysics, GADGET (Springel, 2005) and PHANTOM (Price et al., 2018) are well-established codes for modeling cosmic structure, gas dynamics, and magnetohydrodynamics, offering efficient handling of self-gravity and adaptive time integration. SWIFT (Schaller et al., 2016) provides a scalable, task-based alternative optimized for modern High-Performance Computing (HPC) architectures. In the engineering domain, GPUSPH (Hérault et al., 2010) exploits full GPU acceleration to simulate free-surface and geophysical flows, while SPHinXsys (Zhang et al., 2021) introduces strong fluid-solid coupling and multi-resolution SPH for multi-physics problems. PySPH (Ramachandran et al., 2021) offers a high-level Python interface, making it particularly attractive for method development, educational use, and rapid prototyping. Two particularly versatile and widely adopted environments, LAMMPS, with its SPH module for particle-based continuum mechanics, and DualSPHysics, specialized for weakly compressible and incompressible flows and GPU-accelerated engineering applications, will be discussed in detail in the following subsections. The open-source SPH solvers DualSPHysics and LAMMPS were chosen for their wide adoption in academia and industry, robust community support, and flexibility in implementing new numerical models, making them ideal for methodological enhancements and validation.

2.7.1 LAMMPS code

The LAMMPS code is a code used for particle simulation. It was discovered and developed at Sandia National Laboratories (USA). It was initially employed to study MD problems; however it can also apply for particle simulations following Newton's equations of motion (Ganzenmüller, 2011). The logical diagram of the algorithm followed in the processes related to the SPH method is shown in Appendix A. There is a multitude of commands which are listed with a detailed explanation via the Lammps platform. Among the key principles of the code is the definition of the units and dimensions that the model will be made. It runs on different Windows, Linux, and iOS environments with preference for ease of installation of the SPH library to be done on Linux. The generated script runs through the terminal on the computer and generates different files depending on the command entered by the user. To create animations, users can visualize the output using the program 'OVITO'. More details can be found in Appendix A.

In order to be able to run an SPH simulation there must be at least four structures called pair-particle variables , i.e., density, internal energy E = m-e, and their first derivative with respect to the unit of time. To meet the above requirement a new structure (data structure) was introduced in the code which is identified by the command:

atom_style meso.

which is substituted with the *sph* command in recent LAMMPS versions.

2.7.2 DualSPHysics code

DualSPHysics is an open-source computational tool specifically developed for solving fluid dynamics problems using the SPH. DualSPHysics was created with the objective of leveraging the power of SPH while overcoming some of its computational limitations through efficient parallelization techniques, especially by utilizing GPU computing. The code was developed by a collaboration between the University of Manchester, the

University of Vigo, and other partners, with a strong focus on handling large-scale simulations involving free-surface flows. DualSPHysics is written in C++ for CPU-based computations and incorporates CUDA (Compute Unified Device Architecture) to take advantage of the massively parallel computing capabilities of GPUs. This enables significant speedup for simulations with millions of particles, making it suitable for real-time or high-resolution scenarios. It provides several essential features that make it a powerful tool for researchers and engineers working on complex fluid simulations:

- GPU Acceleration: The most notable feature of DualSPHysics is its ability to run on graphics processing units (GPUs), significantly speeding up simulations. By exploiting the parallel computing capabilities of GPUs, it allows for real-time simulations with a large number of particles, which would be computationally prohibitive using CPU-only methods.
- Support for Fluid-Structure Interaction (FSI): DualSPHysics can simulate the interaction between fluids and solid bodies, making it highly suitable for problems involving floating structures, offshore platforms, and hydraulic machines.
- Complex Boundary Handling: The software provides several techniques to model complex geometries and boundary conditions, such as moving boundaries, solid obstacles, and open boundary conditions.
- Multiphase Flows: The code supports the simulation of multiphase flows, allowing researchers to investigate phenomena like oil spills, sediment transport, or interactions between different fluids.
- Scalability: DualSPHysics is designed for large-scale simulations, with the capability to model millions of particles, which makes it suitable for both academic research and industrial applications where large domains or high-resolution results are necessary.

Before running a simulation, DualSPHysics users typically need to set up the problem domain, define boundary conditions, and generate particles. This is often done using pre-processing tools such as:

- DesignSPHysics: A pre-processing GUI based on FreeCAD that allows users to set up simulations visually. DesignSPHysics simplifies geometry creation, boundary condition application, and particle generation through an intuitive interface, which is especially helpful for non-expert users.
- Gmsh: Although DualSPHysics is a mesh-free method, Gmsh can be used to generate particles and define the initial geometry in certain scenarios. It is particularly useful when complex geometries or initial conditions require more control.

Post-processing is essential for interpreting the simulation results and extracting meaningful data. In addition to Paraview, the following tools are commonly used:

- PartVTK: A utility included with DualSPHysics that converts simulation outputs into VTK format for visualization in Paraview or similar tools.
- MeasureTool: A post-processing tool within DualSPHysics that allows users to measure specific quantities of interest, such as pressure or velocity at locations in the domain over time. This is particularly useful for gathering quantitative results without needing to visualize the entire simulation.
- CSV Data Extraction: Many simulation outputs can be exported as CSV files for easy numerical analysis using spreadsheet programs or scripting in Python, MATLAB, or other data analysis tools.

More information about DualSPHysics code and the visualization tool Paraview (which is the tool used for the visualization purposes of the simulations presented in this study) can be found in Appendix B.

3 Applications of the SPH method-Results

In this chapter we validate the performance and accuracy of the SPH method across various benchmark scenarios involving laminar and turbulent flows in both two-dimensional (2D) and three-dimensional (3D) configurations. Validation is carried out through comparisons against analytical solutions, established numerical benchmarks, and experimental datasets, confirming the capabilities and limitations of the method in diverse practical scenarios. These cases serve to assess the method's ability to capture key physical phenomena, including fluid flow behavior, wave propagation, impact dynamics, and multiphase interactions. Special attention is given to the influence of kernel functions, boundary treatments, and numerical stability in determining the accuracy of SPH simulations.

The SPH method has been widely applied across various fields, demonstrating its effectiveness in solving complex fluid and solid mechanics problems. However, to ensure its accuracy and reliability, extensive validation against experimental data, analytical solutions, and benchmark numerical results are essential.

By systematically comparing SPH results with reference solutions, we highlight both the strengths and limitations of the method, providing insights into its applicability for real-world engineering and scientific problems.

3.1 2D Closed Channel-Laminar flow.

The 2D closed-channel laminar flow serves as a fundamental validation case for the SPH method, assessing its ability to accurately simulate viscous fluid motion in a confined domain. This test is particularly important for evaluating the treatment of boundary conditions, viscosity modeling, and the overall stability of the numerical scheme.

In this scenario, a viscous fluid is enclosed within two parallel solid walls, with an initial velocity profile that evolves over time due to shear forces. The flow is expected to reach a steady-state parabolic velocity distribution, which can be compared to the analytical solution derived from the Navier-Stokes equations.

Key factors analyzed in this test include the accuracy of velocity predictions, pressure distribution, and the influence of kernel functions and particle resolution on the numerical results. The validation of this case demonstrates the SPH method's capability in handling internal flows and modeling viscous effects with high precision. We run these cases in two code SPH code environments LAMMPS and DualSPHysics code.

3.1.1 LAMMPS code

A number of planar Poiseuille-flow channels have been simulated under the effect of an external driving force, F_{ext}. Simulations are verified with the respective analytical solutions. Figure 7a depicts the Poiseuille-flow channel, where the distance between the two plates, h_1, lies within the range and four different channels filled with water particles were constructed and simulated. The contraction channel (Figure 7b) is characterized by the ratio $\frac{h_2}{h_1}$. Simulations are performed for $\frac{h_2}{h_1}$ =10%, 25% and 50% contraction. Similar channels have been studied with sudden expansion at the cross-section (Figure 7c). Ratios are of comparable magnitude, with the key distinction being that one represents a decrease while the other signifies an increase. In the SPH simulations, the computational domain is defined as a simulation box with periodic boundary conditions in the z-direction. The fluid flow is driven by an external force applied in the x-direction to all fluid particles within a specific range.

It is important to note that in LAMMPS particle-based simulations, it is generally preferable to apply an external force F_{ext} individually to each of the N fluid particles. The resulting pressure difference Δp can then be determined using the relationship $\Delta\mathrm{p} = \frac{F_{ext}N}{A}$, where A represents the cross-sectional area.

In SPH simulations, fluid particles are modeled as discrete masses that move according to the flow dynamics. The channel boundaries are defined by stationary SPH particles, which contribute to SPH calculations while maintaining zero velocity. However, these boundary particles still retain their internal energy and local density values. Numerical characteristics of the simulated channels can be found in Table 2.

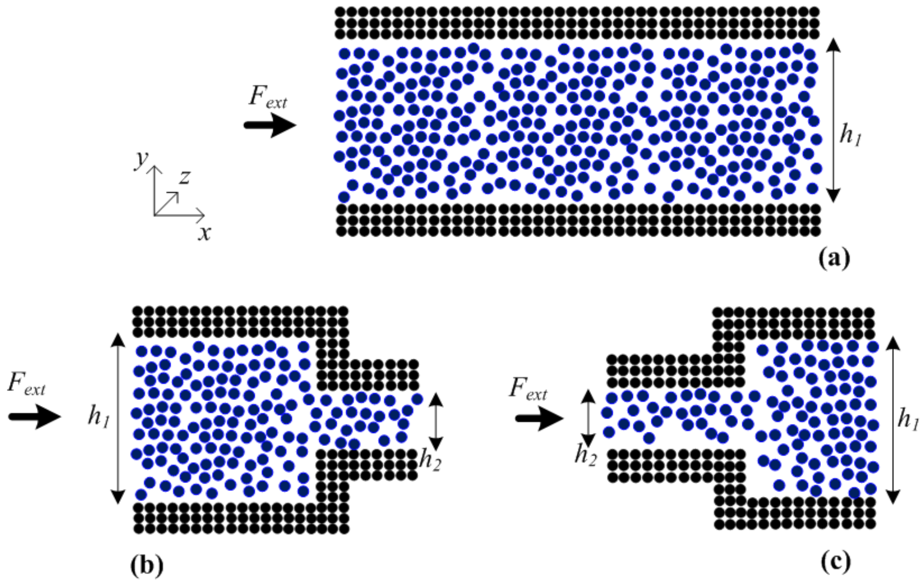


Figure 7. Channel models simulated, a) a planar Poiseuille-flow channel, b) a contraction planar channel and c) an expansion planar channel.

Table 2. Numerical characteristics of the simulated channels.

Channel	h_1 (m)	c_r or e_r	N_f
Poiseuille	4.925×10^{-4}	0%	2418
Poiseuille	7.387×10^{-4}	0%	3658
Poiseuille	8.865×10^{-4}	0%	4402
Poiseuille	9.850×10^{-4}	0%	4836
Contraction	9.850×10^{-4}	10%	4699
Contraction	9.850×10^{-4}	25%	4507
Contraction	9.850×10^{-4}	50%	4191
Expansion	9.850×10^{-4}	10%	4489
Expansion	9.850×10^{-4}	30%	3943
Expansion	9.850×10^{-4}	100%	3029

For all channel cases, fluid density is constant at $\rho = 1000\frac{kg}{m^3}$, dynamic viscosity $\mu = 1 \times 10^{-3} Pas$ and kinematic viscosity $\nu = 1 \times 10^{-6}\frac{m^2}{s}$. The simulation time step is set to Δt=0.5μs. The choice of the time step complies with the CFL criterion for SPH $\Delta t \leq \frac{0.3 \cdot c}{h_{chan}}$ and the viscous condition $\Delta t \leq \frac{0.125 \cdot h_{chan}^2}{\nu}$ (Ellero & Adams, 2012; Korzani et al., 2017).

At the start of the simulation, both water and boundary particles are arranged on an initial lattice within the simulation box. They gradually settle into equilibrium over an initial run of 5×10^6 time steps. This is followed by an additional equilibrium phase of the same duration, during which fluid particles are assigned random velocities to achieve the target temperature. Once equilibrium is established, simulations driven by an external force are conducted, and the computed values are averaged over time. Below the results of this study (Sofos et al., 2022) are presented.

We investigate the impact of the channel's width, h_1, on fluid velocity for 2-D Poiseuille flow (Figure 7a). For all cases shown, an external force of $F_{ext} = 1 \times 10^{-9} N$is used to drive the flow, corresponding to a pressure gradient $\frac{dp}{dx}$ of $4.5 \frac{Pa}{m}$. The Reynolds number is defined as $Re = \frac{\bar{u} h_1}{\nu}$, where $\bar{u}$ is the cross-sectional mean velocity. The analytical solution for the velocity profile in planar Poiseuille flow is given by (Liakopoulos, 2019)

$$u = -\frac{dp}{dx} \frac{\left(\frac{h_1}{2}\right)^2}{2\mu} \left(1 - \frac{y^2}{\left(\frac{h_1}{2}\right)^2}\right) \quad (3\text{-}1)$$

where h_1 is the distance between the parallel plates and the x-axis lies in the mid-plane. Greater channel width leads to greater velocity values, as expected. Calculated results fit well with the corresponding analytical solutions (see Figure 8).

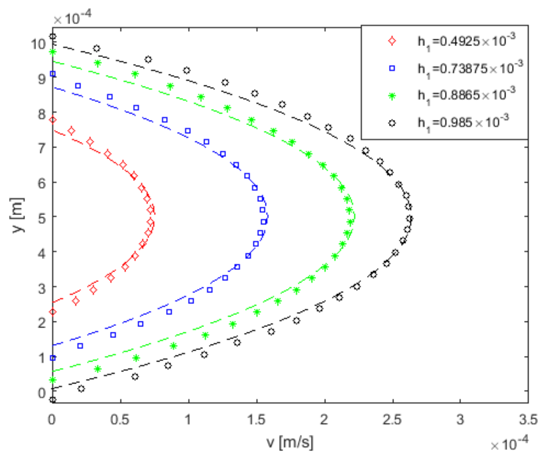
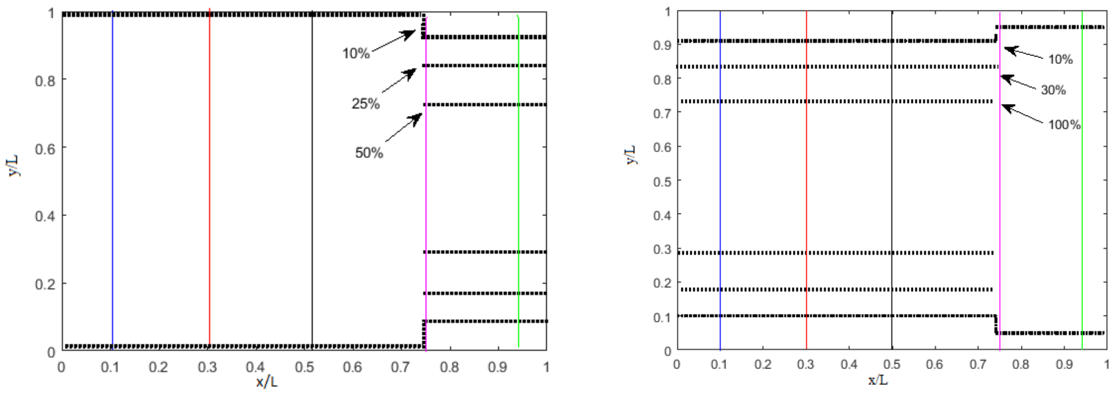


Figure 8. Poiseuille flow velocity profiles for various channel widths, h1=(0.4925, 0.73875, 0.8865, 0.985) x 10^{-3} m, where Re= (0.2235, 0.7318, 1.2757, 1.6883), respectively. Lines are the respective analytical solutions.

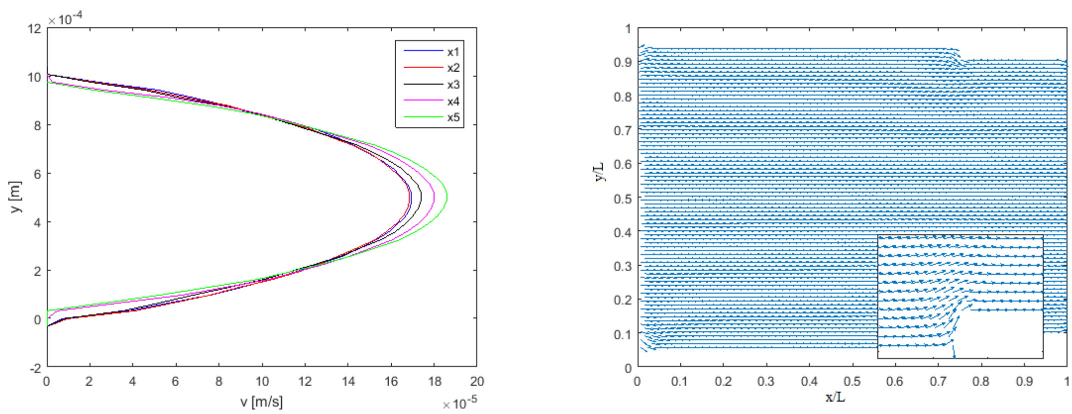
Next, we investigate fluid flows in planar channels which exhibit sudden contraction at some cross-section *x=const*. The severity of the contraction is quantified by the ratio c_r. Points along the *x*-axis where velocity profiles are calculated are shown in Figure 9a for the contracted and Figure9b for the expanded channels, respectively, (x_1, x_2, x_3, x_4, x_5) = (0.1,0.3,0.51,0.75,0.94) m, respectively. As there are two different cross-sections for every expanded and contracted channel, we calculate two Reynolds numbers, Re_l for the left side and Re_r for the right side.



(a) (b)

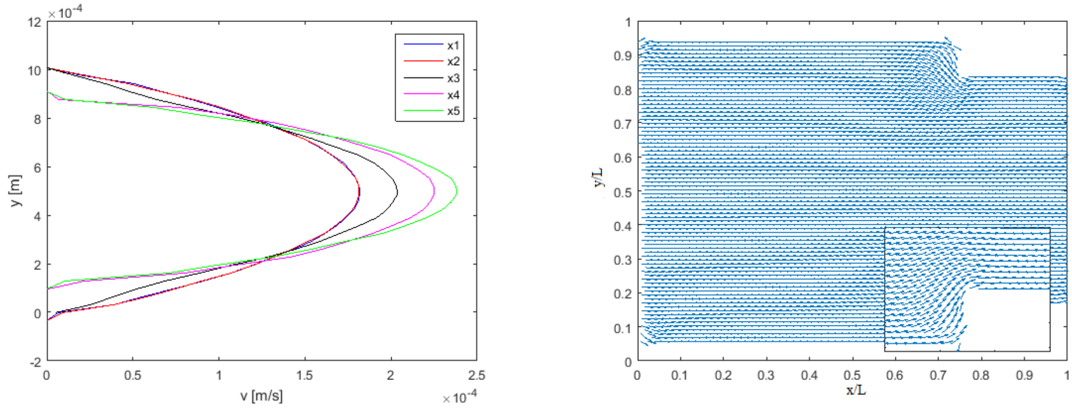
Figure 9. Points on the x-axis (scaled values) for the extraction of velocity profiles along the channels. a) contraction channel and d) expanded channel. Points are referred to as x1-x5, from left to right.

Results of the velocity profiles and particles trajectories for c_r = 10, 25 , 50 % are depicted in Figures 10, 11, 12, respectively.



(a) (b)

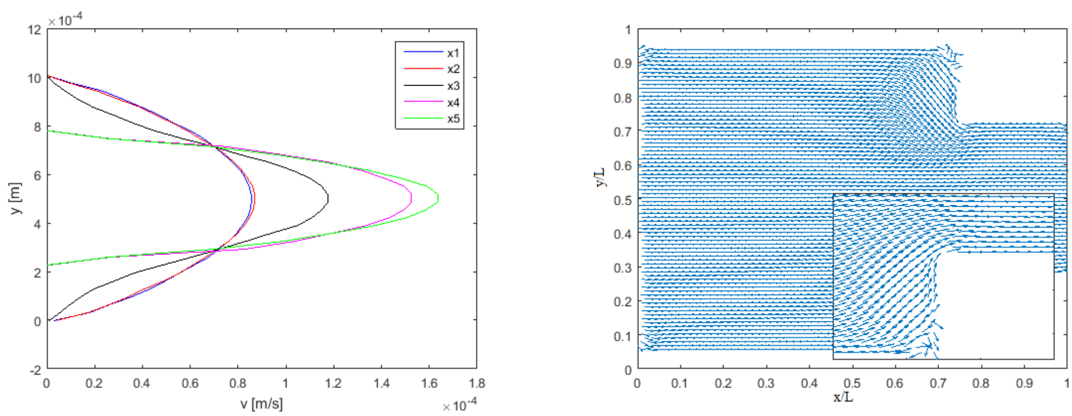
Figure 10. Velocity distribution for the c_r=10% contracted channel, a) velocity profiles (dimensional values) and b) particle trajectories (quiver plot, scale =1.0) with zoomed corner view. Re_l=0.107 and Re_r=0.102.



(a) (b)

Figure 11. Velocity distribution for the c_r =25% contracted channel. a) velocity profiles (dimensional values) and b) particle trajectories (quiver plot, scale =1.0) with zoomed corner view. Re_l=0.115 and Re_r=0.11.

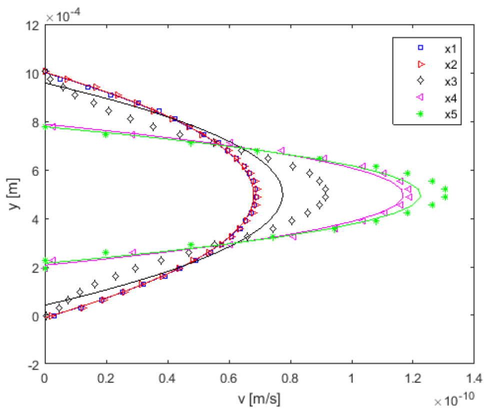
For the c_r=50% contraction channel, Figure 12a, we observe that smaller velocity values are obtained in all cross-sections, compared to the previous contraction channels in Figures 10 and 11. Due to the sudden boundary change at the contraction region, a non-parabolic velocity profile is obtained at cross-section x_3. The computed velocity field (trajectories) in Figure 10b is symmetrical to the channel mid-plane. No recirculation region is created at the contraction corner, and this is attributed to the small value of the Reynolds number and the small cross-section decrease. As expected, small errors are reported in velocity values/directions near the boundary corners.



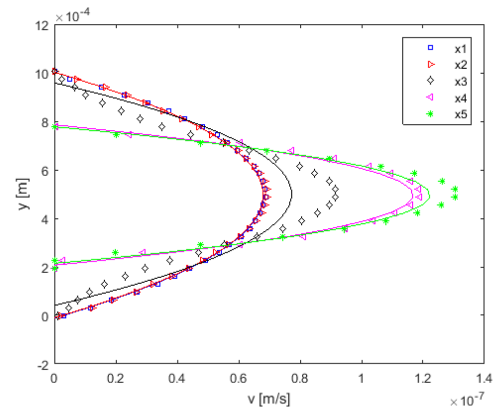
(a) (b)

Figure 12. Velocity distribution for the c_r =50% contracted channel. a) velocity profiles (dimensional values) and b) particle trajectories (quiver plot, scale =1.0) with zoomed corner view. Re_l=0.0271 and Re_r=0.0345.

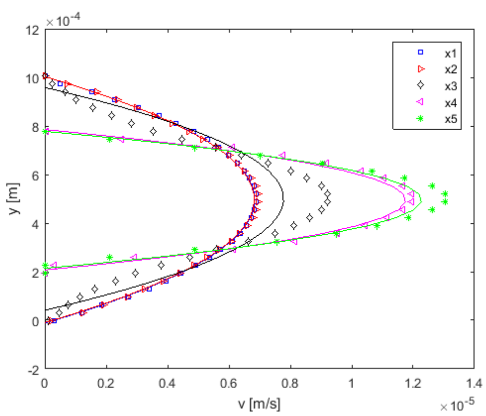
Figures 13(a-d) depict the effect of the externally applied force, which corresponds to the pressure difference, on fluid velocity in a channel with a 50% contraction. As expected, an increase in the external force (F_{ext}) leads to higher fluid velocity. The velocity profiles at cross-sections x_1 and x_2, located after the channel inlet, align well with parabolic fits, consistent with the characteristics of a steady, fully developed flow where h_1 remains constant. A similar parabolic shape is observed at cross-section x_4, which is positioned immediately after the contraction. However, near the contraction at cross-section x_3 and near the channel outlet at cross-section x_5, the velocity profiles deviate significantly from a parabolic shape.



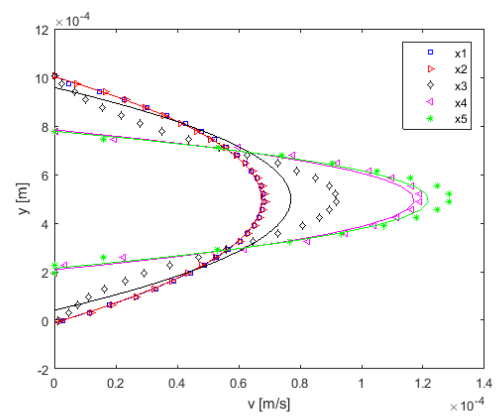
(a)



(b)



(c)



(d)

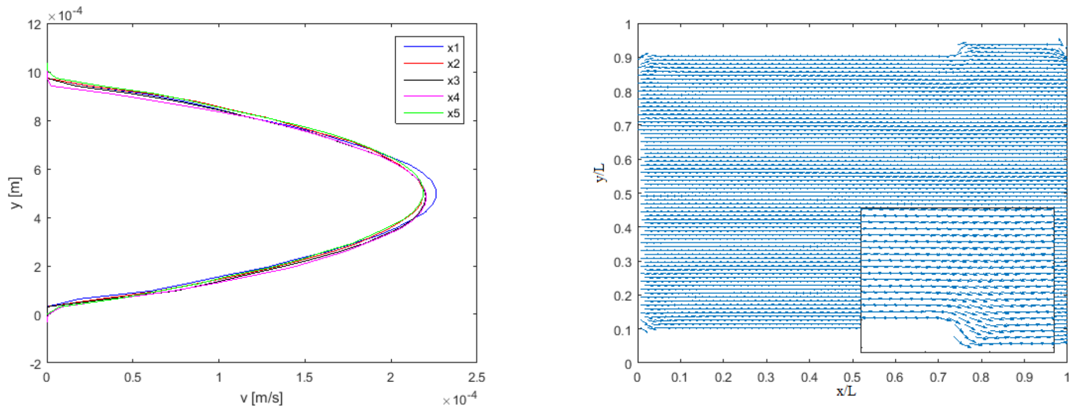
Figure 13. Velocity values for various magnitudes of F_{ext} for the 50% contracted channel, a) $F_{ext} \leq 1 \times 10^{-15} N$, b) $F_{ext} \leq 1 \times 10^{-12} N$, c) $F_{ext} \leq 1 \times 10^{-10} N$ and d) $F_{ext} \leq 1 \times 10^{-9} N$. Lines are 2nd order parabolic fits.

The flow characteristics for the e_r=10% expansion are depicted in Figures 14(a-b). No significant changes are observed in the shape and values of the velocity profiles in cross-sections x_2-x_5. As in the $c_r = 10\%$ contraction channel, they resemble the respective profiles shown before for the Poiseuille flow (Figure 8). However, in Figure 14a, a slightly higher velocity value emerges near the channel inlet (cross-section x_1). On the contrary, for the $c_r = 10\%$ contraction channel (Figure 10a) shown before, higher velocity values are taken near the channel outlet.

As fluid particles pass through the contraction line, they are confined to a smaller region, causing their velocities to increase as they adjust to their new positions. Conversely, after the expansion line in a widened channel, particles decelerate as they redistribute within the larger flow area. The velocity field, which visualizes particle trajectories (Fig. 14b), exhibits symmetry along the channel's midplane, with minor inaccuracies observed near the expansion corners.

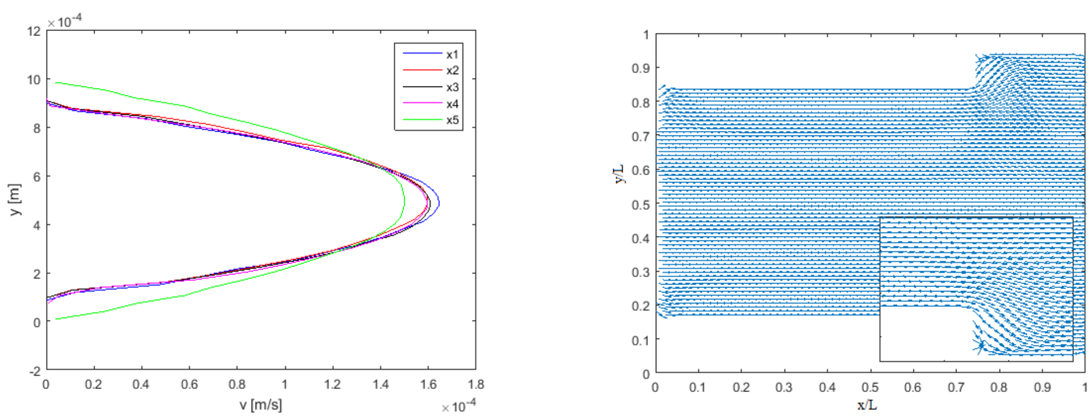
For the $e_r = 30\%$ expansion, the corresponding velocity profiles are presented in Figures 15a and 15b. The highest velocity values are observed along the centerline near the channel inlet (cross-section x_1), similar to the 10% expansion case. In this scenario, where the channel expansion is more pronounced, the deceleration of fluid particles as they adjust to a wider flow region becomes even more evident. This effect is particularly noticeable in the velocity profile at cross-section x_5 in Figure 15a and is even more pronounced in the $e_r = 100\ \%$ expansion channel (Fig. 16a), where the profile deviates significantly from a parabolic shape.

The velocity field (Fig. 15b) remains symmetrical about the channel midplane both before and after the expansion. Minor discrepancies are observed at the corners following the expansion, which is also the case for the $e_r = 100\ \%$ expansion (Fig. 16b). Additionally, no vortices are present, and the flow stream expands smoothly after the sudden widening of the channel.



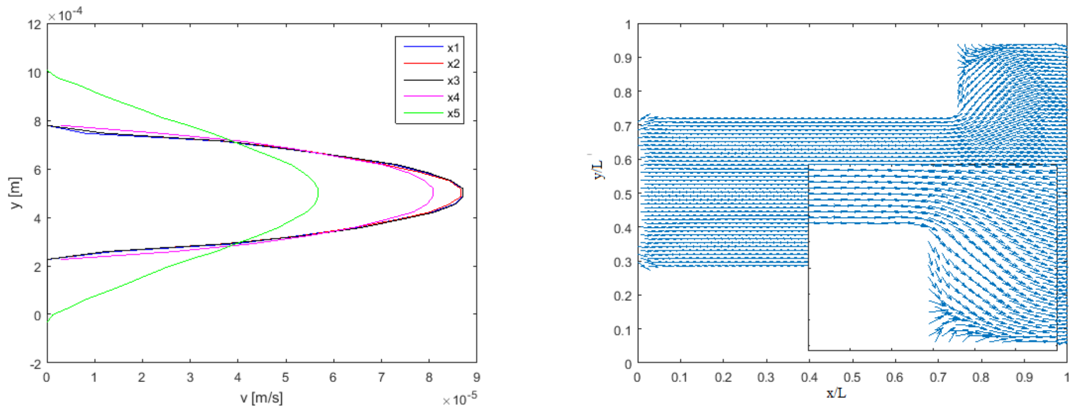
(a) (b)

Figure 14. Velocity distribution for the e_r=10% expansion channel, a) velocity profiles (dimensional values) and c) particle trajectories (quiver plot, scale =1.0) with zoomed corner view. Re_l=0.112 and Re_r=0.124.



(a) (b)

Figure 15. Velocity distribution for the $e_r = 30\%$ expansion channel. a) velocity profiles (dimensional values) and c) particle trajectories (quiver plot, scale =1.0) with zoomed corner view. Re_l=0.0745 and Re_r=0.0936.



(a) (b)

Figure 16. Velocity distribution for the e_r=100% expansion channel, a) velocity profiles along *x*-direction (scaled values), b) velocity profiles (dimensional values), c) particle trajectories (quiver plot, scale =1.0) and d) velocity vector field - zoomed view at the expansion corner. Re_l=0.0177 and Re_r=0.0256.

3.1.2 DualSPHysics code

Using the same approach, a study was conducted on similar cases employing the DualSPHysics code (Chatzoglou et al., 2023). The first set of simulations involves four scenarios of water flow through closed 2D microchannels, with the nominal water density assumed to be 1000 kg/m³. In the simulations with the DualSPHysics code, the liquid is modelled as weakly compressible, with small local variations in water density (about 0.1%). The parameters for the baseline simulations, boundary conditions (BCs), and the computational time required on an RTX 3060 GPU are summarized in Table 3. Here, DBC refers to dynamic boundary conditions, Np represents the number of fluid particles, dp is the initial interparticle distance, and ν_0 denotes the kinematic viscosity.

Table 3. Model parameters of 2D baseline simulations.

Case	Poiseuille	Developing Flow	Sudden Expansion	Expansion/Contraction
N_p	10,000	10,060	10,340	15,300
dp (m)	2.00×10^{-5}	2.00×10^{-5}	2.98×10^{-5}	2.98×10^{-5}
ν_0 (m^2/s)	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}
BCs at solid wall	DBC	DBC	DBC	DBC
BCs in x-direction	Periodic	Inlet/outlet	Inlet/outlet	Periodic
Force per unit mass (m/s^2)	$g_x = 0.001$	$g_x = 0.0$	$g_x = 0.0$	$g_x = 0.001$
Time Integration	Verlet	Verlet	Symplectic	Symplectic
Simulation time (s)	5	5	20	30
GPU time (min)	2.1	2.5	10	8

3.1.1. Constant Cross Section Microchannel

Fully Developed Flow Model

In the first preliminary test case, the fluid is confined between two very large parallel solid plates (Figure 17a). The gap between the plates is H = 2h = 0.985 mm. Periodic PBCs are applied in streamwise (x-axis) and in spanwise (z-axis) (Figure 17b). This computational setup corresponds to the well-known Poiseuille flow. Zero velocity is applied at t = 0 everywhere in the flow field in the baseline simulation. After a certain simulation time, the flow reaches steady conditions. In further simulations, the same solution is reached independently of the initial flow conditions considered.

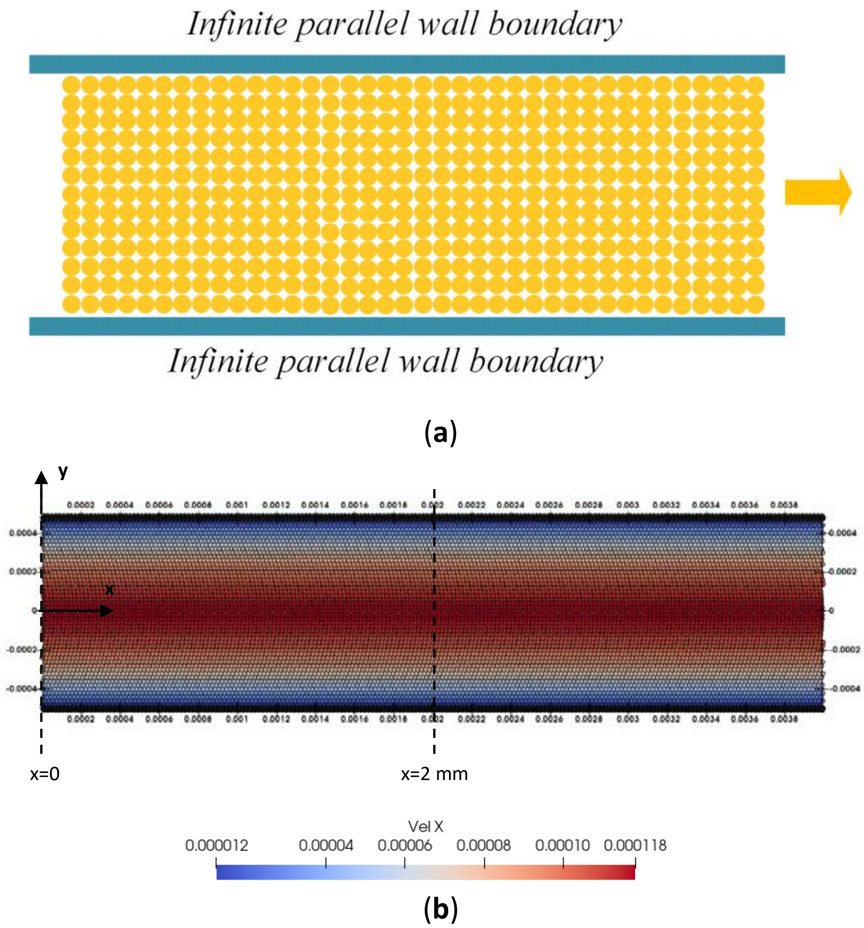


Figure 17. (a) Longitudinal illustration of the Poiseuille case. (b) Two-dimensional constant cross section channel. Geometry and coordinate system in 2D simulations.

The x-velocity component, computed using the SPH method, is compared to the analytical velocity profile in Figure 18. For easy reference, we provide the analytical solution to the N–S equations for laminar Poiseuille flow here (Liakopoulos, 2019). The exact velocity field is

$$u = -\frac{Gh^2}{2\mu}\left(1 - \frac{y^2}{h^2}\right), v = 0, w = 0 \qquad (3\text{-}2)$$

where the y-axis is in the wall–normal direction, μ is the dynamic viscosity of the fluid, and 2h is the gap between the two plates. Eq. (3-2) is valid for a coordinate system with the x-axis in the mid-plane of the configuration. The driving pressure gradient is denoted by $G = \frac{\partial P}{\partial x}$ and is a negative constant. It is noted that, instead of a pressure gradient, one can, equivalently, apply a body force of ρg_x to drive the flow (see Table 3). For values up to g_x = 0.1 m/s^2, the velocity profile is in very good agreement with the analytical solution and the linear behavior is obvious. Values greater than 0.1 m/s^2 have not been tested.

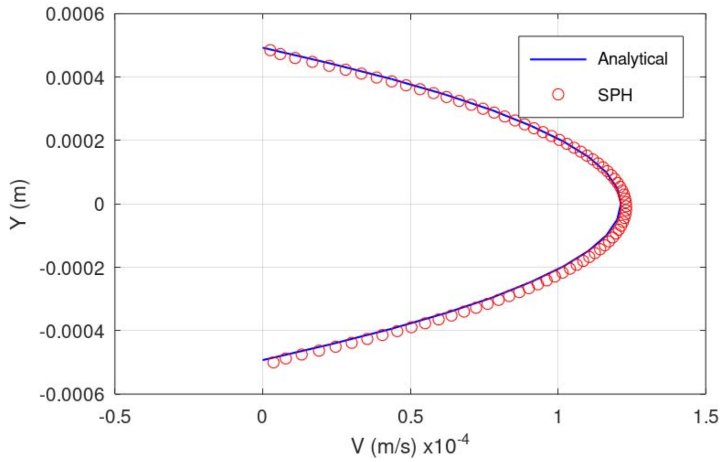


Figure 18. Poiseuille flow. Computed velocity profile at x = 2 mm. Comparison with the analytical profile (Eq. (3-2)).

The size of the gap between the two parallel plates as well as the bulk velocity of the water are small and, consequently, in all simulations, the flow Reynolds number is low. As a result, the nonlinear terms of the N–S equations are negligible. The linear response of the flow is verified by doubling the driving force per unit mass using a number of numerical experiments.

Wall Shear Stress and Friction Factor

The shear stress at the solid walls of channels is of great practical importance across all scales: nano, micro, and macro (Liakopoulos, 2019), since it is directly related to the pressure drop along the channel and, consequently, to the power required for fluid transport. For a Newtonian fluid, the wall shear stress is calculated as

$$\tau_w = \mu \frac{du}{dy}\bigg|_{wall} \tag{3-3}$$

and, for the coordinate system described in the context of Eq. 3-3, $|\tau_w| = |G|h = \mu\, u_{max}\frac{2}{h}$. In Table 4, we summarize the comparison between the computed velocity derivative at the wall in the baseline simulation and its exact analytical value, and the relative difference obtained is 2.04% for both boundary conditions. By increasing the driving force per unit mass, we further simulated Poiseuille flow at different values of the Reynolds number. In Figure 19, we present the relation between the friction factor, f, and the flow Reynolds number as predicted by the SPH simulation. The definitions adopted here are

$$f = \frac{\tau_w}{\frac{1}{8}\rho\bar{V}^2} = \frac{(-\frac{\partial p}{\partial x})D_h}{\rho\frac{\bar{V}^2}{2}}, Re = \frac{\bar{V}D_h}{\nu_0}, D_h = \frac{4A}{P} = 2H = 4h, \tag{3-4}$$

where D_h is the hydraulic diameter and $\bar{V}$ is the mean velocity at the cross-section.
Table 4. Two-dimensional constant cross-section channel. Comparison between SPH results with analytical solution for the derivative of x-velocity component.

Baseline Simulation	SPH	Analytical	Rel. Diff. (%)
Periodic BCs	0.48	0.49	2.04%
Inlet/outlet BC with buffer zones	0.48	0.49	2.04%

The least squares fit of the computed values of the friction factor deliver f = 95.9/Re (see Figure 19), which is in excellent agreement with the exact value of the Poiseuille number, $Po = f \times Re = 96 =$ const.

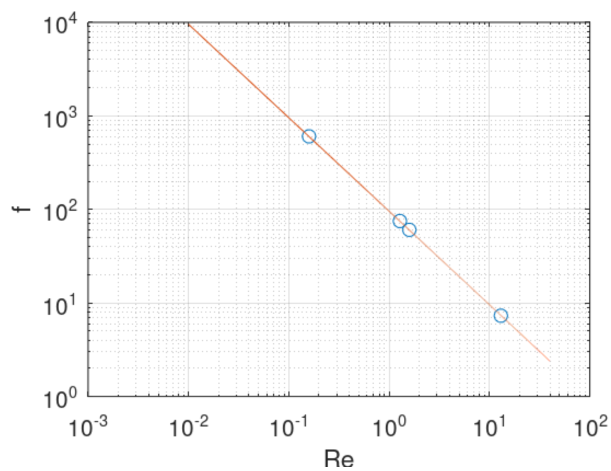


Figure 19. Two-dimensional Poiseuille flow. Least square fits the f vs. Re relation based on SPH simulations.

Developing Flow Model

Next, we examine the developing laminar flow between two very large parallel plates. Since the developing length of the flow is not known beforehand, PBCs in the x-direction are not suitable and are instead replaced with inlet/outlet boundary conditions. In this setup, two buffer zones are created during the pre-processing step. At the inlet buffer zone, uniform velocity and pressure are applied, both of which are constant in time and uniform in space. These buffer zones interact with the fluid particles through ghost nodes, transferring all necessary information to the fluid particles at the inflow and outflow. In the baseline simulation, the uniform velocity at the inlet is set to 8.075×10^{-5} m/s. The initial conditions represent water in equilibrium. Once the flow reaches fully developed conditions, the SPH results align excellently with the exact solution for fully developed Poiseuille flow (see Figure 20a).

To investigate the model's response to an increased flow rate, we doubled the inlet velocity. As a result, the maximum velocity, mean velocity, flow rate, and Reynolds number all doubled. The SPH results for the fully developed flow again show excellent agreement with the analytical Poiseuille profile (see Figure 20b).

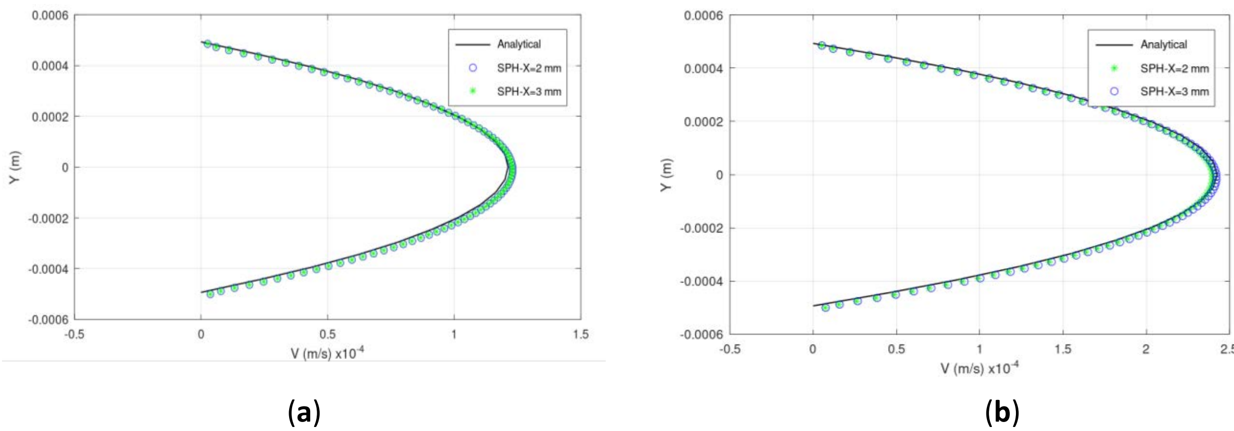


Figure 20. Developing flow model. Computed velocity profile at x = 2mm and x = 3 mm. Comparison with the analytical profile (Eq. 3-2). (a) $\mathrm{V_{inlet}} = 8.075 \times 10^{-5}\,\mathrm{m/s}$. (b)$\mathrm{V_{inlet}} = 1.605 \times 10^{-4}\,\mathrm{m/s}$.

Two-Dimensional Sudden Expansion

In this numerical experiment, a sudden increase in the linear dimensions of the channel cross-section is introduced. The distance between the two solid walls increases suddenly from 0.4925 mm to 0.985 mm at x = 4 mm. The geometry of the channel and the length of each portion of the channel are shown in Figure 5. Boundary conditions in the streamwise direction are applied using an inlet buffer zone with constant-in-time and uniform-in-space velocity and free outflow conditions at the outlet of the computational domain (see Figure 21). The inlet velocity is set equal to 3.125 × 10−5 m/s, while the initial velocity conditions are set equal to zero $u = v = w = 0$ everywhere in the flow field.

Figures 5 and 6 show the particles and the trajectory lines at t = 20 s when the flow reaches steady flow conditions. Well-defined vortices are formed immediately after the channel's sudden expansion. Although the Reynolds number is small, the flow separates because of the sharp 90° corner and reattaches after a length slightly larger than the "step" height (see Figure 22).

Velocity profiles in two different longitudinal positions (Figure 23) exhibit the expected parabolic behavior with the max velocity values to be located at the center of the closed channel and velocity values tend to zero, close to the boundary (solid–fluid interface). Some inaccuracies occurred, mainly due to the treatment of fluid–solid interface BCs since strict no-slip conditions have not been implemented yet in the DualSPHysics version used for this study (Dominguez et al., 2022). Inaccuracies are more apparent in the narrower cross-section. This is mainly due to the separation distance between the plates in combination with the larger velocity.

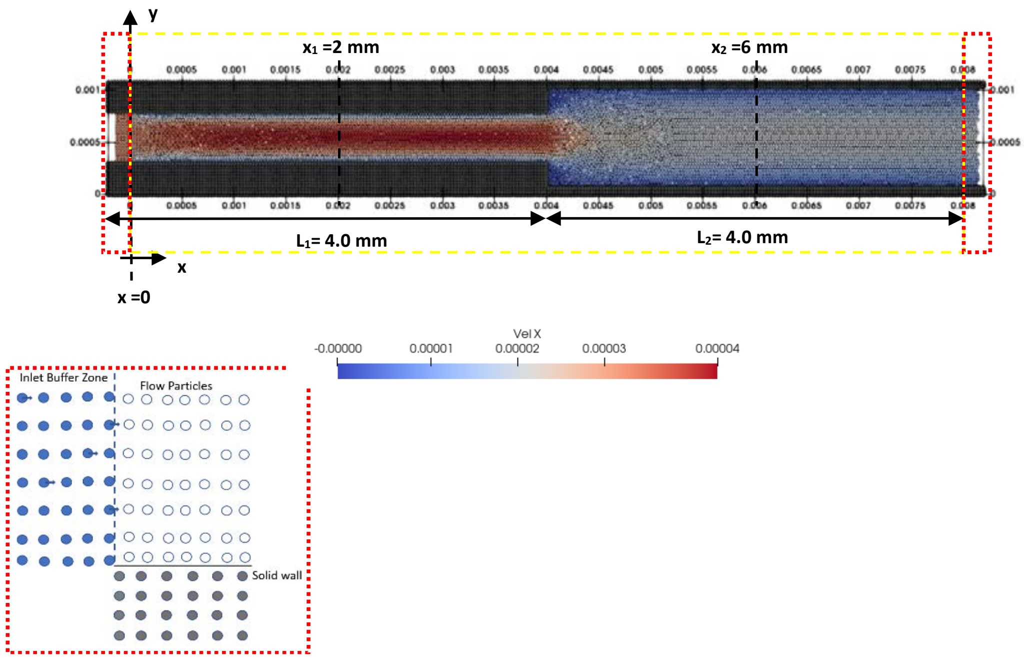


Figure 21. Two-dimensional sudden expansion. Overall geometry and computational domain. Flow field snapshot at t =20 s.

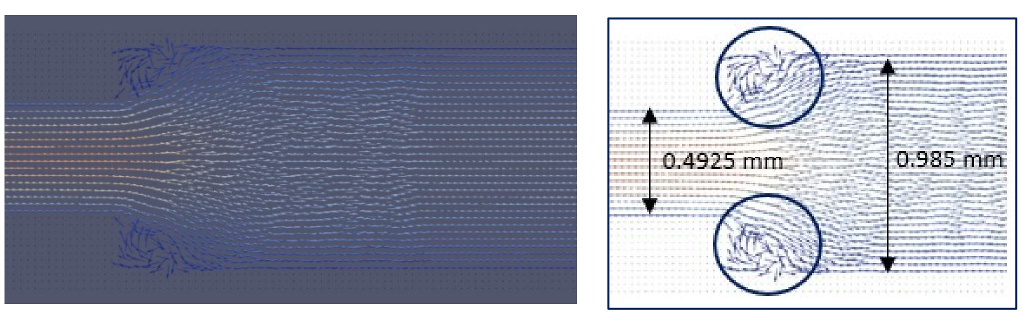


Figure 22. Two-dimensional sudden expansion. Flow field snapshot and trajectory lines at t = 20 s. Arrows demonstrate velocity direction and are colored by velocity values (not scaled).

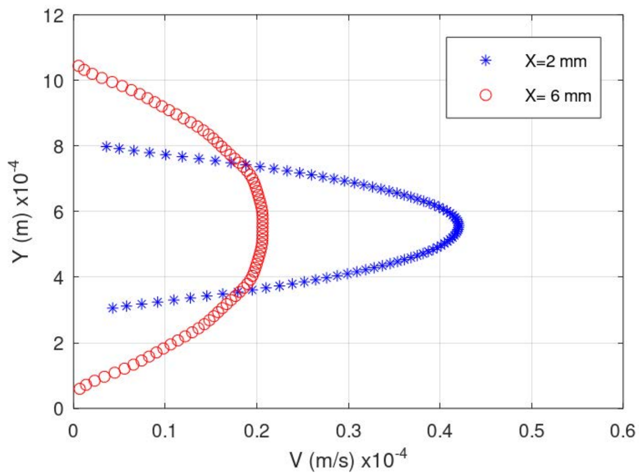


Figure 23. Two-dimensional sudden expansion. Velocity profiles at longitudinal positions x_1 = 4 mm and x_2 = 6 mm. At x1, $\bar{V}_1 = 2.89 \times 10^{-5}\ \mathrm{m/s}, (Q_1 = 1.42 \times 10^{-11}\ \mathrm{m^3/s})$, while at x2, $\bar{V}_2 = 1.43 \times 10^{-5}\ \mathrm{m/s}, (\mathrm{Q}_2 = 1.41 \times 10^{-11}\ \mathrm{m^3/s})$.

Two-Dimensional Sudden Expansion/Contraction

In the final 2D case considered, the distance between the solid walls suddenly increases from 0.4925 mm to 0.985 mm at x=0 and then decreases back to 0.4925 m at x=3.6 mm (see Figure 24). The objective in this case is to model the flow in a long channel with periodic expansions and contractions. The SPH model outlined in this subsection solves the Navier–Stokes equation for a single module of the channel, as illustrated in Figure 24. PBCs are applied in the streamwise direction, and the initial velocity field is set to zero. The flow eventually reaches a steady state after some time.

Figure 25 provides a detailed view of the streamline/pathline (trajectory) pattern in the central part of the computational domain. The SPH model effectively captures the curvature of the streamlines and the formation of vortices near the corners.

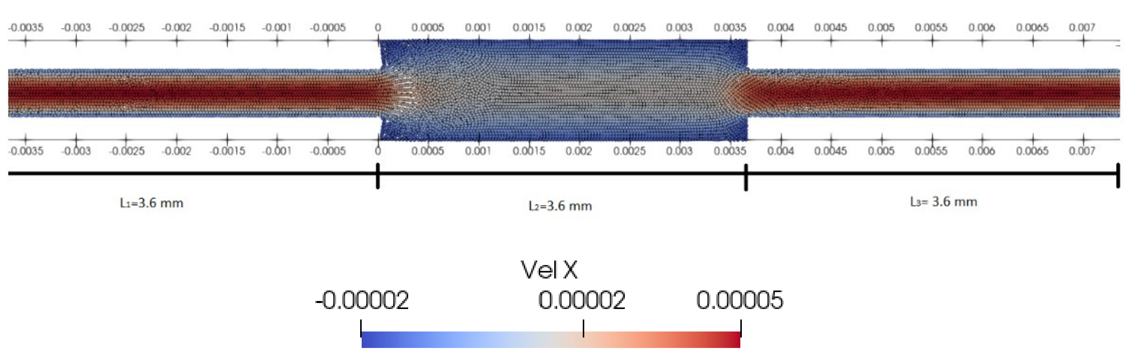


Figure 24. Two-dimensional sudden expansion/contraction. Overall geometry and dimensions of the computational domain. Flow field snapshot at t = 30 s.

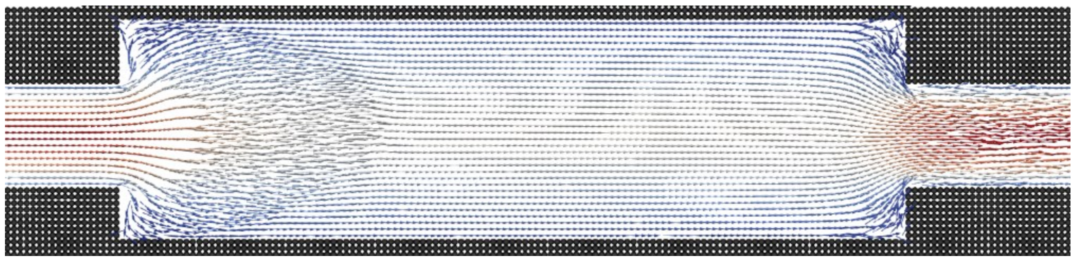


Figure 25. Two-dimensional sudden expansion/contraction. Trajectory lines (not to scaled) in the expanded section of the channel at t = 30 s.

The computed velocity profiles exhibit the expected parabolic behavior at cross sections located far from the sudden expansions and contractions. In Figure 26, the velocity profiles at longitudinal positions $x_1 = -2$ mm, $x_2 = 2$ mm, and $x_3 = 6$ mm are shown. Corresponding cross-sectional average velocities are $\bar{\boldsymbol{V}}_1 = 1.5 \times 10^{-5}\,\mathrm{m/s} = \bar{\boldsymbol{V}}_3, \bar{\boldsymbol{V}}_2 = 3 \times 10^{-5}\,\mathrm{m/s}$, and $\mathrm{Q}_1 = \mathrm{Q}_2 = \mathrm{Q}_3 = 7.38 \times 10^{-12}\,(\mathrm{m^3/s}) = 7.38 \times 10^{-2}\,\mathrm{\mu L/s}$.

The principle of mass conservation is fully upheld, meaning that no fluid particles leak through the channel walls, which is a common issue in purely particle-based numerical methods.

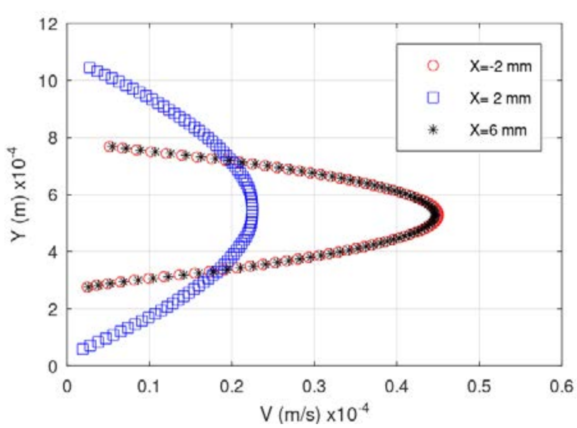


Figure 26. Two-dimensional sudden expansion/contraction. Velocity profiles at selected longitudinal positions $x_1 = -2$ mm, $x_2 = 2$ mm, and $x_3 = 6$ mm. Corresponding cross-sectional average (mean) velocities $\bar{V}_1 = 1.5 \times 10^{-5}\,m/s = \bar{V}_3, \bar{V}_2 = 3 \times 10^{-5}\,m/$

s, and volume flowrates are $\mathrm{Q}_1 = \mathrm{Q}_2 = \mathrm{Q}_3 = 7.38 \times 10^{-12}\ (\mathrm{m}^3/\mathrm{s}) = 7.38 \times 10^{-2}\ \mu\mathrm{L/s}$.

3.2 3D Closed Channel-Laminar flow

Similar test cases are simulated in three dimensions to evaluate the accuracy of the SPH method in a more realistic context. 3D microchannels have significant applications in industries such as biomedical engineering, chemical processing, and electronics, where laminar flow is often critical. For example, microfluidic devices used in lab-on-a-chip technology rely on precise laminar flow to control the mixing, separation, and analysis of fluids. Additionally, 3D microchannels are employed in compact heat exchangers for efficient heat transfer in systems like electronics cooling and automotive radiators. Laminar flow also plays a key role in microreactors used in chemical engineering, where controlled flow conditions are essential for efficient chemical reactions and process optimization. By simulating these applications in three dimensions, we can better assess the SPH method's performance and its relevance to real-world scenarios. The two code environments (LAMMPS and DualSPHysics) are tested also in this group of problems.

3.2.1 Lammps code

The 3D channels are enclosed by wall particles, with fluid flowing from the inlet to the outlet. The numerical parameters of the system are provided in Table 5. All channels feature square cross-sections (Figure 27). As in previous cases, the contraction channel is defined by the ratio c_r, while the expansion channel is characterized by the expansion ratio e_r. These ratios describe the change in height between the inlet and outlet of the channels.

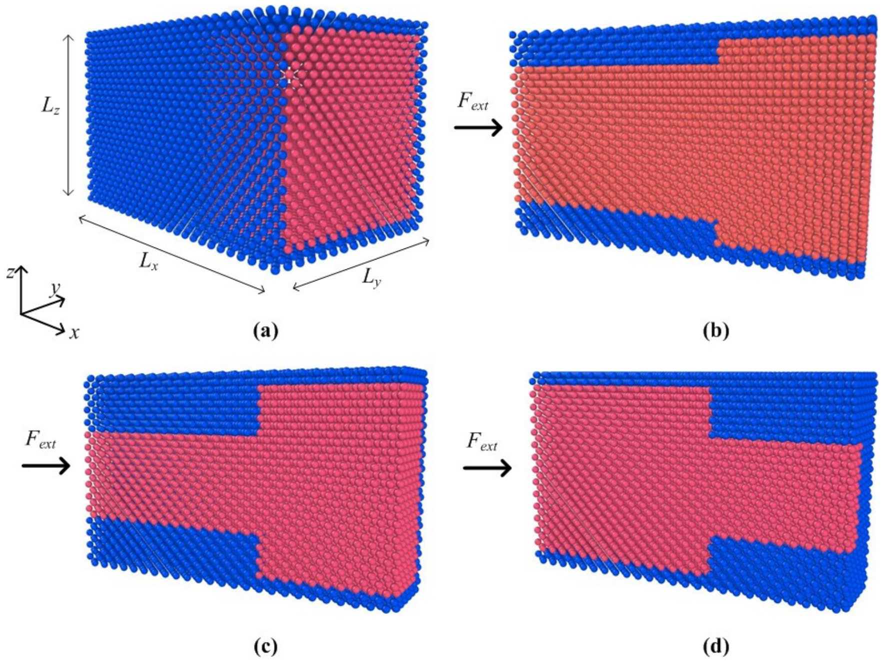


Figure 27. Solid and fluid particles for 3-D channels with sudden expansion/contraction. (a) 3D view of expansion channels, (b) slice view of 30% expansion, (c) slice view of 100% expansion, and (d) slice view of 50% contraction.

Table 5. Numerical characteristics of the 3-D channels. L_i represents the simulation box dimensions in the i-direction (where i=x, y, z).

Channel	Lx (m)	Ly (m)	Lz (m)	h (m)	cr or er	Nf
Constant cross-section	18.12×10^{-4}	11.04×10^{-4}	11.04×10^{-4}	9.85×10^{-4}	0%	18,589
Constant cross-section	18.12×10^{-4}	11.04×10^{-4}	11.04×10^{-4}	4.925×10^{-4}	0%	4,998
Contraction	18.12×10^{-4}	11.04×10^{-4}	11.04×10^{-4}	9.85×10^{-4}	50%	13,259
Expansion	18.12×10^{-4}	11.04×10^{-4}	11.04×10^{-4}	9.85×10^{-4}	30%	13,490
Expansion	18.12×10^{-4}	11.04×10^{-4}	11.04×10^{-4}	9.85×10^{-4}	100%	10,328

We first present the calculated velocity field in a straight channel of constant square cross-section (Figs. 28(a-b)). As expected, increased velocity values are found near the channel centerline and values decrease as we approach the solid walls. Figure 28(c) compares the velocity profiles for h=0.985x10-3 m and 0.4925 x10-3 m channels when the external driving force is $F_{ext} = 1\times10^{-3}N$. Both profiles fit on a 2nd order polynomial. Figure 28(d) presents, as surface plot u(y,z), the velocity distribution on the yz-plane at x=0.000475 m, as shown in Fig. 28(b). In all 3-D results the definition of the Reynolds number is based on hydraulic diameter $D_h = 4R_H = 4A/P$, where A is the cross-sectional area and P the wetted perimeter.

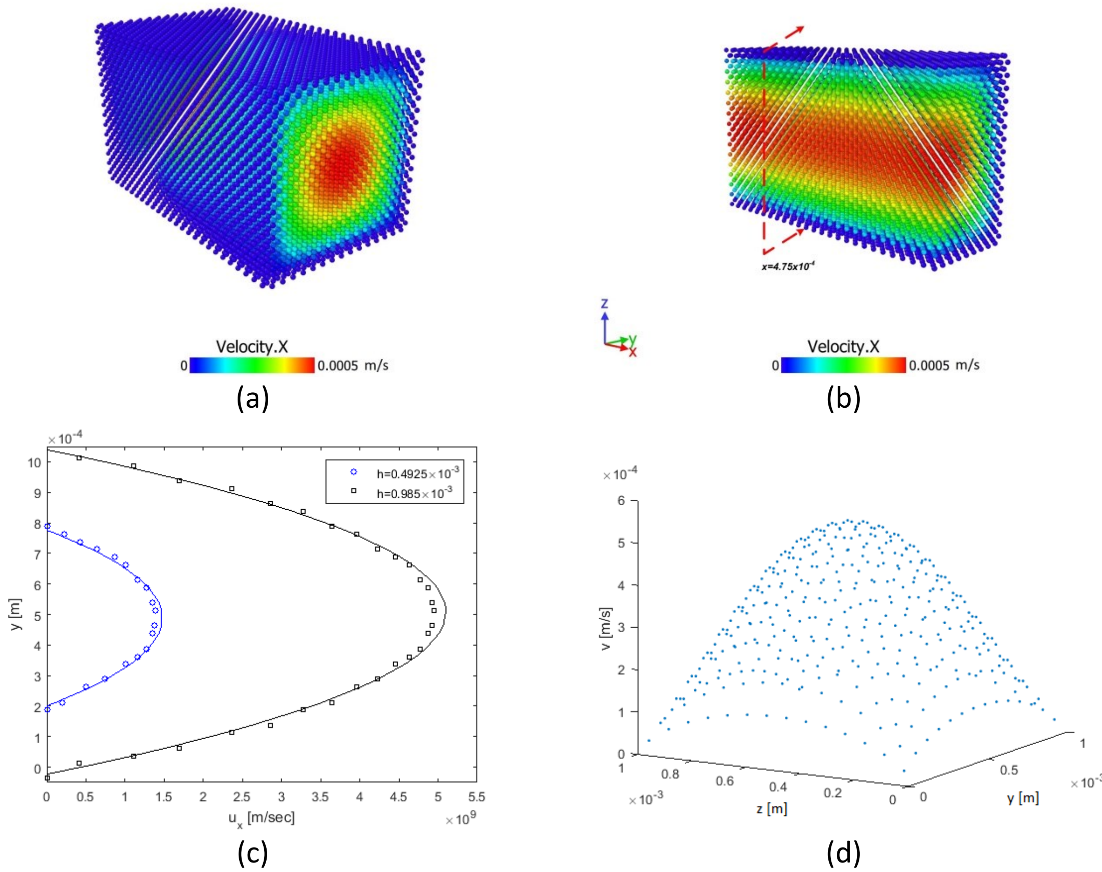


Figure 28. Velocity field for a rectangular channel, a) 3D view, b) slice view, c) velocity profiles and d) x-velocity distribution on yz-plane for the h=0.985x10-3 m channel, Re=0.281.

In Figures 29(a-b), the computed velocity field is presented for the c_r=50% contraction channel. The field is symmetrical with respect to the channel centerline. No recirculation regions or vena contracta appear, due to the low Reynolds number of the flow (creeping flow). The values of velocity increase as the flow moves towards the contraction region, satisfying the conservation of mass principle.

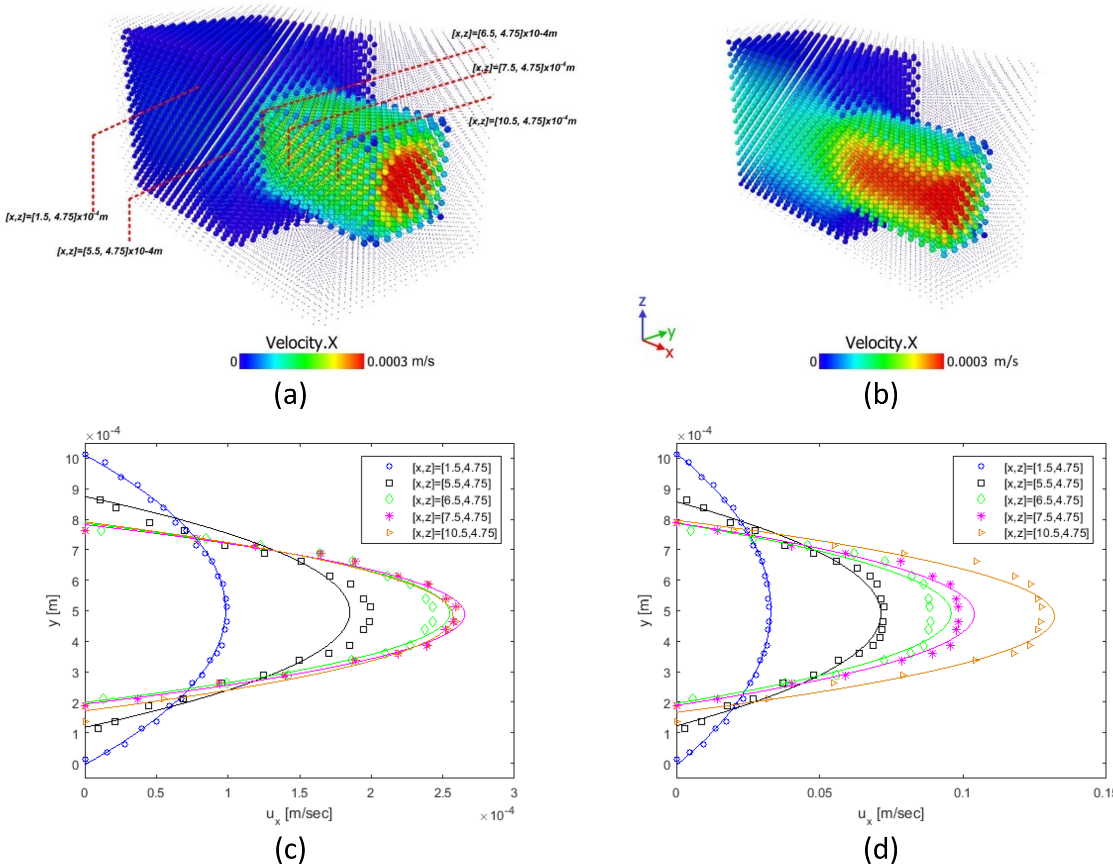


Figure 29. Velocity field for the 50% contraction channel, a) 3D view, b) slice view and velocity profiles for c) $F_{ext} = 1 \times 10^{-3}N$, Re_l=0.0486 and Re_r=0.0408, and d) $F_{ext} = 1N$, Rel=18.7 and Rer=16.3.

This behavior is also apparent when we compare the velocity profiles in Figs. 29(c-d), for $F_{ext} = 1 \times 10^{-3}N$ and $F_{ext} = 1N$, respectively. Here, velocity distributions are presented as functions of y at various line segments ([x,z] planes shown in Fig. 29(a)). Near the channel inlet, at line segment $[x, z] = [1.5, 4.75] \times 10^{-4}$, the profile fits well on 2nd order polynomial at steady flow conditions. Parabolic shape is also observed near the channel outlet. However, in some of our simulations the flow has not reached fully developed conditions after the contraction or expansion. We believe that this is not a problem since our interest lies within the low-Re flow characteristics near the sudden change of the cross-sectional area. Velocity values increase at the outlet (mean velocity: $\bar{u}_x = 15.64 \times 10^{-5} m/s$) compared to inlet values (mean velocity: $\bar{u}_y = 6.34 \times 10^{-5} m/s$). Near the contraction, on the line segment $[x, z] = [5.5, 4.75] \times 10^{-4}$, the profile is far from parabolic shape, as also observed for the 2-D cases (see Fig. 12).

For $F_{ext} = 1N$ (Figure 29d) different velocity values are observed right after the contraction; velocity increases as we approach the channel outlet. However, in this case, all profiles, even at the contraction corners, fit well on 2nd order polynomials.

For the e_r=30% expansion channel, the velocity field and profiles are presented in Figure 30. The field is naturally weaker near the walls compared to the velocity values at the centerline. The respective velocity profiles are shown in Figs. 30(c-d). At first, for Re_l=0.159 and Re_r=0.118 (Fig. 30c), the velocity profiles present lower values as the fluid moves from the narrower side to the expanded side of the channel (flow from left to right, as shown in Figure 27b), and, in parallel, the velocity profiles become more "suppressed". For Re_l=55.4 and Re_r=47.3, when the external driving force has been increased to $F_{ext} = 1N$ (Fig. 30d), we notice of course that all velocities have increased, and, moreover, there are smaller differences between the maximum values of the respective profiles, compared to the $F_{ext} = 1 \times 10^{-3}N$ case.

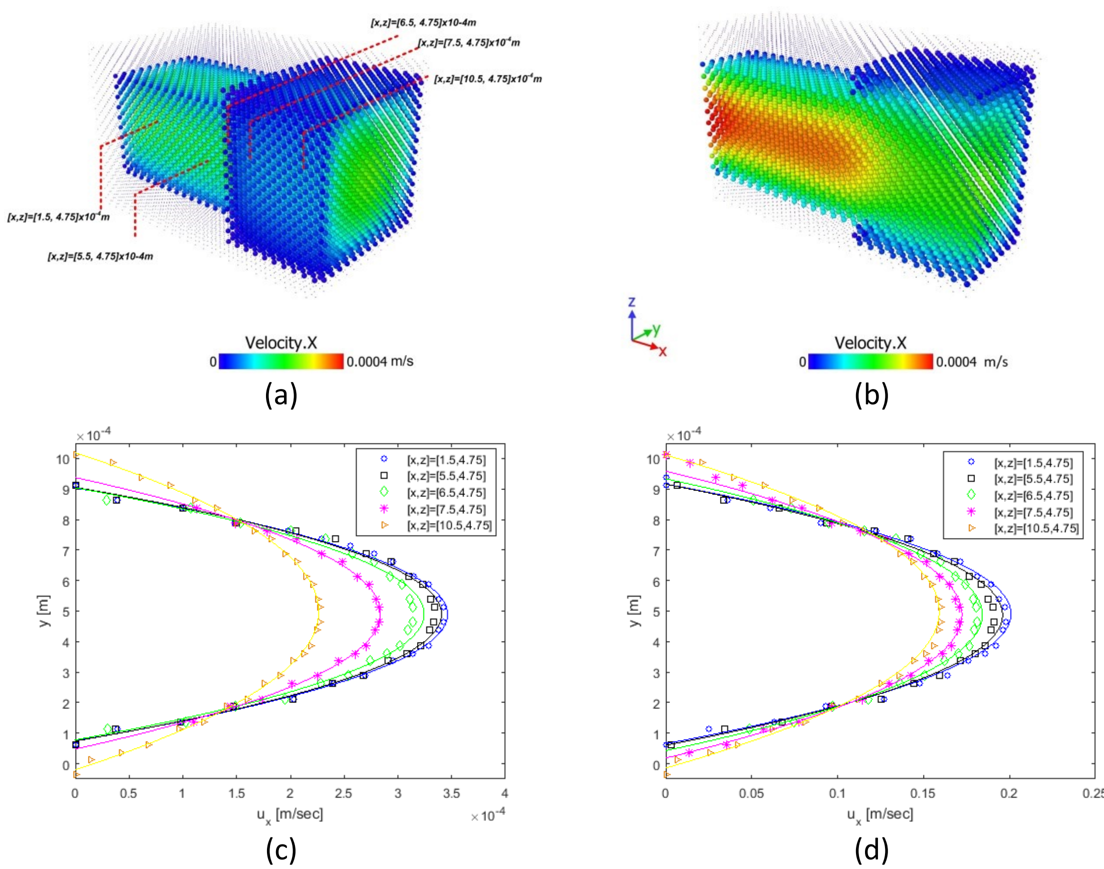


Figure 30. Velocity field for the 30% expansion, a) 3D view, b) slice view and velocity profiles for c) $F_{ext} = 1 \times 10^{-3}N$, Re_l=0.159 and Re_r=0.118, and d) $F_{ext} = 1N$, Re_l=55.4 and Re_r=47.3.

For the e_r=100% expansion channel, the isosurfaces of the velocity field are presented in Figs. 31(a-b) in color-coded plots. The field is weaker in the expanded region, and the particle color near the expansion walls indicate that the fluid is almost immobilized there. The representative velocity profiles shown in Figs. 31(c-d) are lower and wider near the channel inlet and change to higher values and narrower shape as the fluid moves from left to right.

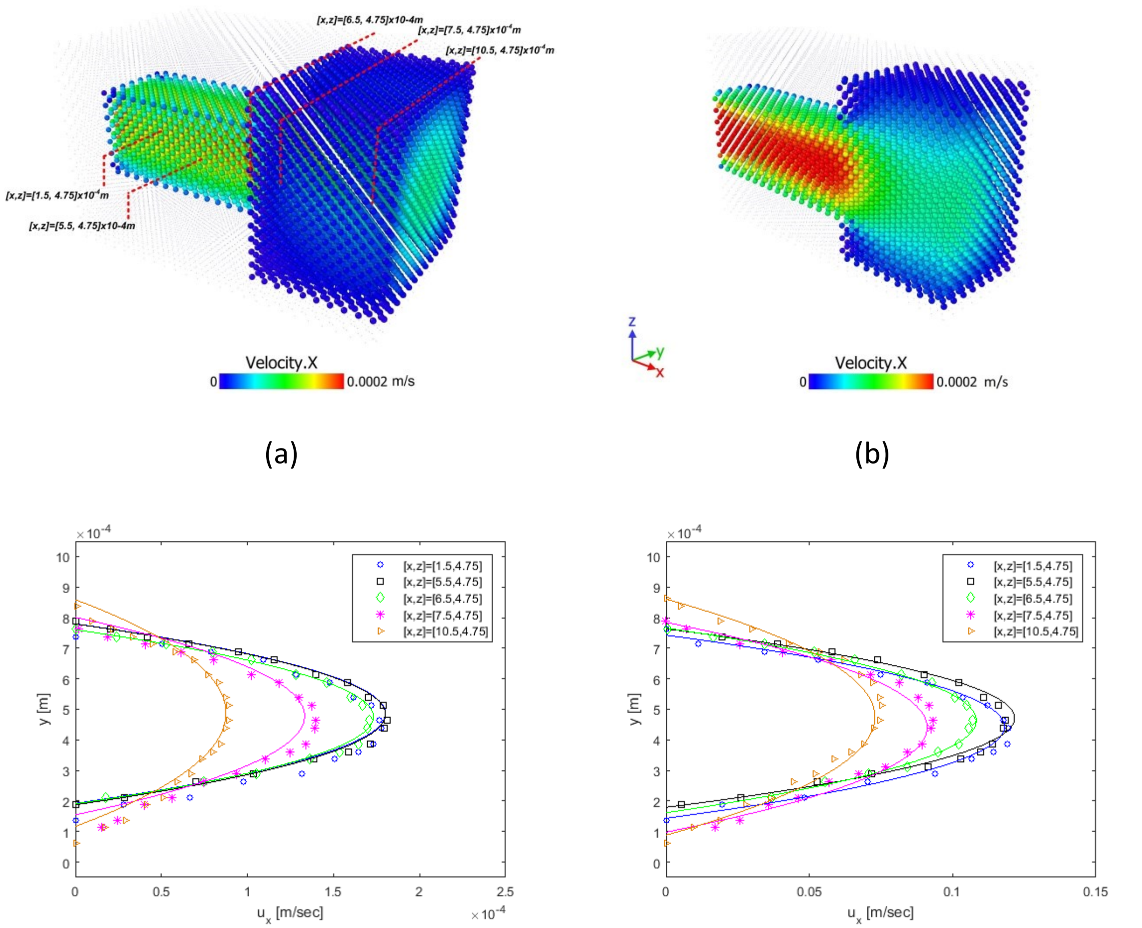


Figure 31. Velocity field for the 100% expansion a) 3D view, b) slice view and velocity profiles for c) $F_{ext} = 1 \times 10^{-3}N$, Re$_l$=0.0615 and Re$_r$=0.034, and d) $F_{ext} = 1N$, Re$_l$=30 and Re$_r$=19.7.

3.2.2 DualSPHysics code

In this section, we present the results of SPH simulations for flows in straight micro ducts with a square cross-section (Chatzoglou et al., 2023). Transverse and longitudinal sections of the studied microchannels are shown in Figure 32. It is important to note that in the 3D SPH simulations, the x-axis is aligned with the lower wall, while the z-axis is directed normal to both the upper and lower walls. The y-axis is positioned in the spanwise direction. Physical and computational parameters, BCs, and the time integration methods used in the baseline simulations are provided in Table 6.

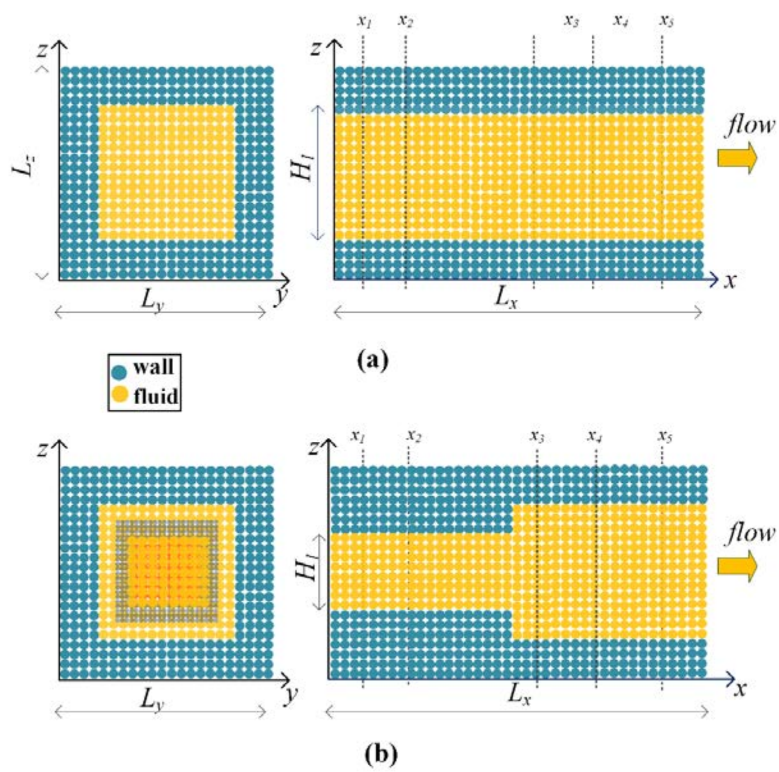


Figure 32. Geometry of the square-microchannel test cases used to validate the 3-D SPH model: (a) straight 3D channel Ly = Lz = 1.104 mm. (b) Channel with sudden expansion at the cross-section. Five solid-particle layers forming the walls. Axes are oriented with x along the bottom wall, y span-wise and z normal to the walls.

The first 3D case considered consists of a constant cross-section long microchannel with $\mathrm{Ly} = \mathrm{Lz} = 1.104\ \mathrm{mm}$ (see Figure 32a). The fluid occupies a net area of 0.985 mm × 0.985 mm while the length of the computational domain in the x-direction is set equal to 3.6 mm. The flow is considered fully developed so that periodic BCs are applied in the x-direction. DBCs are imposed on solid walls. A number of initial conditions for the velocity field were tested (zero, uniform in space, etc.), complying with the conservation of mass principle. The magnitude of the streamwise velocity component (obtained using SPH) is shown in Figure 33 using a color-coded plot.

An exact infinite series solution of laminar flow in a straight channel of a rectangular cross-section can be found in the literature (White, 2006). Specifically, the velocity field and the volume flowrate are

$$u(y,z) = \frac{16a^2}{\mu\pi^3}\left(-\frac{\partial P}{\partial x}\right)\sum_{i=1,3,5..}^{\infty}(-1)^{(i-1)/2}\left[1-\frac{cosh(i\pi z/2a)}{cosh(i\pi b/2a)}\right]\times\frac{cos\,(\frac{i\pi y}{2a})}{i^3} \tag{3-5}$$

$$Q = \frac{4b\alpha^3}{3\mu}\left(-\frac{\partial P}{\partial x}\right)\left[1-\frac{192a}{\pi^5 b}\sum_{i=1,3,5..}^{\infty}\frac{\tanh\,(\frac{i\pi b}{2a})}{i^5}\right] \tag{3-6}$$

In Eqs (3-5 and 3-6), α and b represent the half-lengths of the rectangular cross-section along the y-axis and z-axis, respectively. For the cases considered in this study, α = b. Additionally, in Eq. 3-5, the x-axis is aligned with the duct's axis of symmetry. The SPH numerical solution is compared with the analytical solution by examining the x-velocity distribution in Figure 34 and the velocity profiles in Figure 35. The comparison shows excellent agreement between the two solutions.

Table 6. Parameters of 3D baseline simulations.

Case	Const. C.S. (A)	Const. C.S. (B)	Sudden Expansion	Expan./Contr.
Np	216,480	308,788	581,040	897,334
dp (m)	2.98×10^{-5}	2.98×10^{-5}	2.98×10^{-5}	2.98×10^{-5}
ν_0 (m^2/s)	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}
BCs at solid walls	DBC	DBC	DBC	DBC
BCs in x-direction	Periodic	Inlet/outlet	Inlet/outlet	Periodic
Force per unit mass (m/s^2)	$g_x = 0.001$	$g_x = 0.0$	$g_x = 0.0$	$g_x = 0.001$
Time integration	Symplectic	Symplectic	Symplectic	Symplectic
T (s)	20	50	60	50
GPU time (min)	13.6	24.6	38.85	51

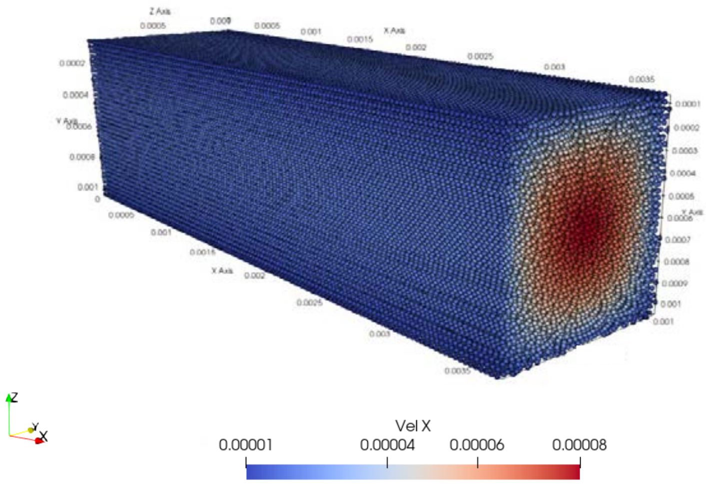


Figure 33. Straight 3-D constant cross-section channel. PBCs. x-velocity component (steady flow, t = 20 s).

The computed volume flow rate (discharge) closely matches the analytical value from Eq. 3-6, with a relative error of 3%. This level of accuracy is achieved using the computational parameters listed in Table 6 under the column labelled "constant C.S. (A)." Increasing the number of particles in the simulation reduces the relative error, provided that the associated parameters are appropriately adjusted.

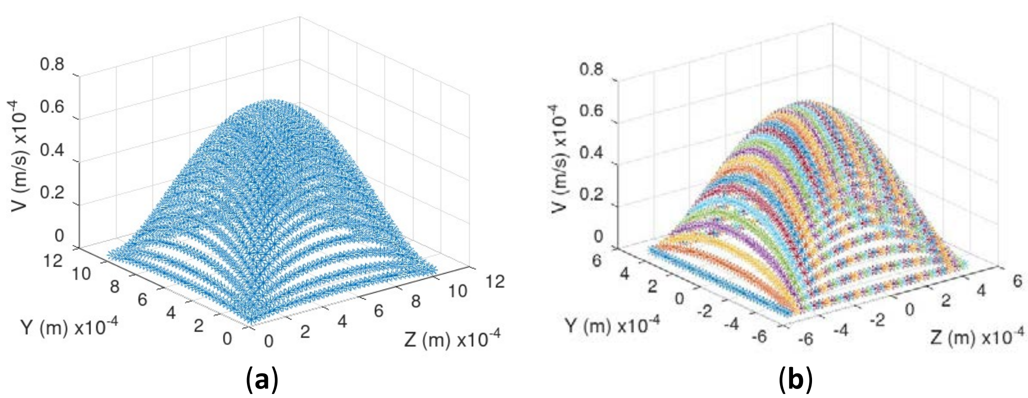


Figure 34. Straight 3-D constant cross-section microchannel. Periodic BCs. x-velocity component distribution u (Y, Z): (a) SPH-computed profile at x = 2 mm and (b) analytical solution (Eq. (3-5)).

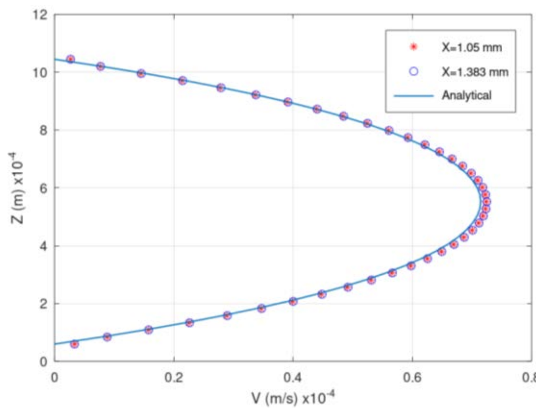


Figure 35. Three-dimensional constant cross-section channel. Periodic BCs. SPH velocity profile $\mathrm{u}\ (\mathrm{Z}, \mathrm{Y} = 0.4925\ \mathrm{mm})$ at x = 1.05 mm and x = 1.383 mm and analytical solution velocity profile $\mathrm{u}\ (\mathrm{z}, \mathrm{y} = 0)$, where $\mathrm{z} = \mathrm{Z} - 0.4925 \times 10^{-3}\ (\mathrm{m}), \mathrm{y} = \mathrm{Y} - 0.4925 \times 10^{-3}\ (\mathrm{m})$.

Poiseuille Number, the Relation f vs. Re

For the design of microdevices, energy loss can be calculated using the relation f vs. Re. By changing the driving force per unit mass, we simulated the flow at various values of the Reynolds number. In Figure 36, we present the relation between the friction factor f and the Reynolds number based on cross-sectional average velocity and hydraulic diameter D_h. In the case of a square duct, the hydraulic diameter D_h is equal to the square side length.

Least squares fit of the computed data provides the correlation $f = 57.4/\mathrm{Re}$ (see Figure 36), which is in very good agreement with the exact value for the Poiseuille number for the square duct, $\mathrm{Po} = f \times \mathrm{Re} = 56.9 = \mathrm{const.}$ (Liakopoulos, 2019).

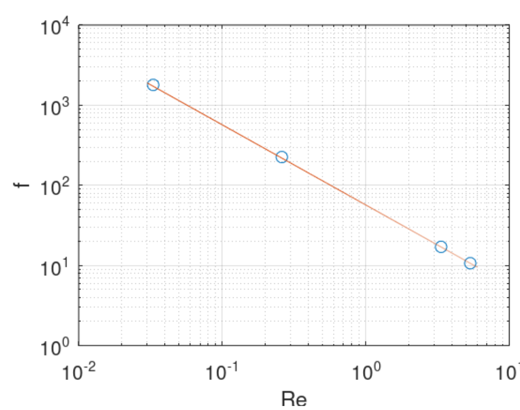


Figure 36. Three-dimensional constant cross-section. Periodic BCs. Least square fits the f vs. Re relation based on SPH simulations.

Developing Flow in a Square Micro duct of Constant Cross-Sectional Area

To assess the SPH performance in simulating 3D developing flow, we replaced the PBCs with inlet/outlet buffer zones. As the inlet condition, we applied constant-in-time and uniform-in-space velocity distribution, $V_{inlet} = 3.4 \times 10^{-5}$ m/s.

The flow reaches steady state conditions after a certain time, and fully developed flow is established at approximately x = 1 mm. The developing length (L_{dev}) is very short and well within the length of the computational domain in the streamwise direction (Figure 37).

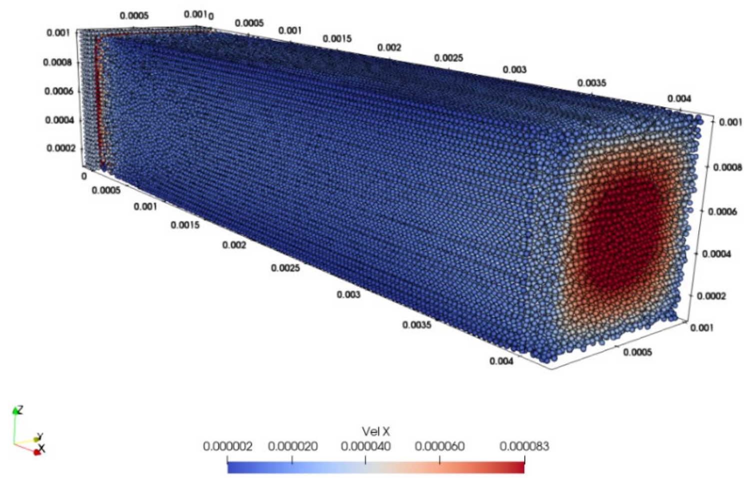


Figure 37. Three-dimensional constant cross-section microchannel. Developing flow. x-velocity component (steady flow, t = 50 s).

The development of the flow profile in the vertical mid-plane (Y = 0.4925 mm) is shown in Figure 38 while the 3D plot of the velocity distribution at x = 1.05 mm is depicted in Figure 39.

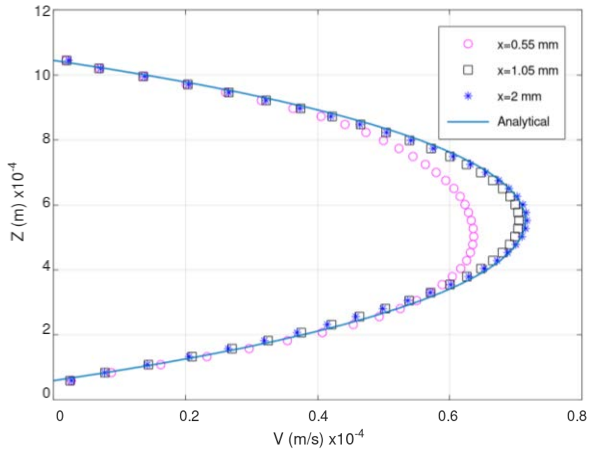


Figure 38. Three-dimensional constant cross-section microchannel. Developing flow. SPH-computed velocity profile development at the vertical mid-plane (Y = 0.4925 mm).

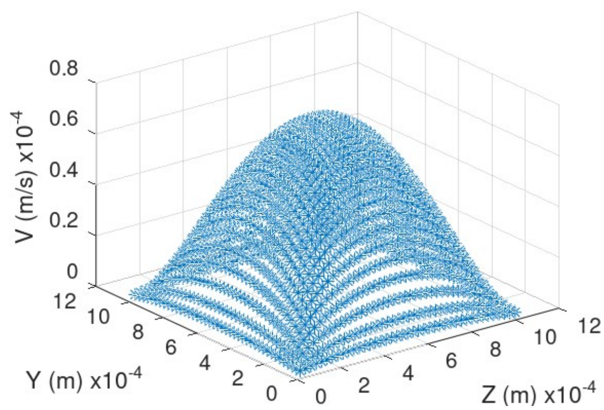


Figure 39. Three-dimensional constant cross-section microchannel. Developing flow. Velocity distribution u (Y, Z) at cross-section x = 2 mm in the fully developed region.

The computed mean velocity at x = 1.05 mm equals $\bar{V}_{\mathrm{sph}} = 3.3 \times 10^{-5}\ \mathrm{m/s}$, showing a relative error of 3% when compared to the analytical solution. Discharge: $\mathrm{Q_{SPH}} = 3.2 \times 10^{-11}\ \mathrm{m^3/s} = 3.2 \times 10^{-2}\ \mathrm{\mu L/s}$. We note that the use of inlet/outlet BCs demanded more computational time, mainly due to the longer time that the flow needs to reach steady state conditions.

In a sudden expansion, it is assumed at x = 4 mm (see Figure 40). For $\mathrm{x} < \mathrm{x}_{exp}$, the net cross-section area that the fluid occupies is $\mathrm{L_y} \times \mathrm{L_z} = 2.426 \times 10^{-7}\ \mathrm{m^2} \approx 0.24\ \mathrm{mm^2}$. For $\mathrm{x} > \mathrm{x}_{exp}$ the net cross section that the fluid occupies is $\mathrm{L_y} \times \mathrm{L_z} = 9.702 \times 10^{-7}\ \mathrm{m^2} \approx 0.97\ \mathrm{mm^2}$. For this case, there are two additional issues that need to be addressed: the selection of the inlet velocity value and the number of particle layers that form the solid boundaries (walls). To hold the fluid particles within the cross-section, we apply five layers of solid particles to form the walls. The solid particle arrangement has the same interparticle distance as that of the initial fluid particle configuration.

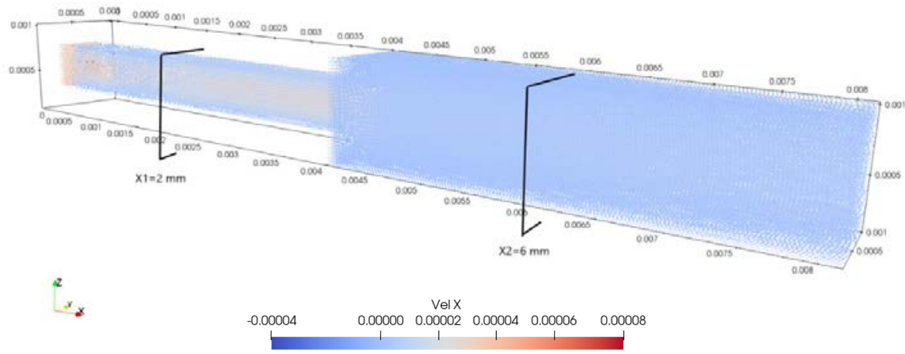
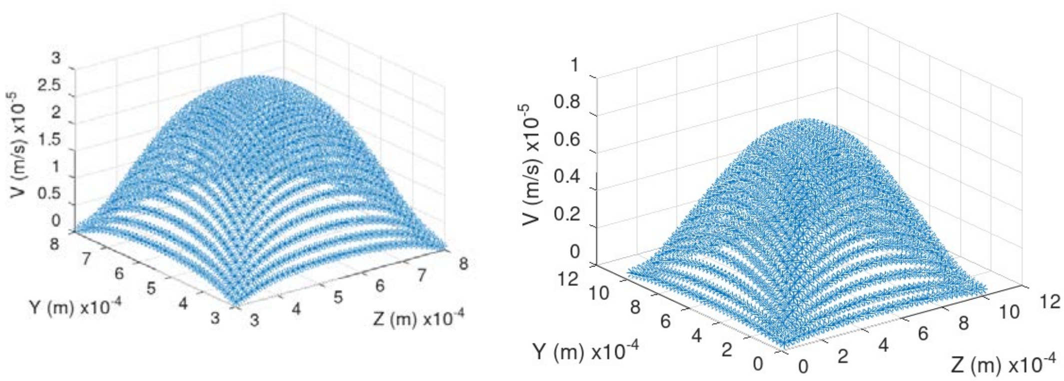


Figure 40. Three-dimensional sudden expansion. x-velocity component (steady flow, *t* = 60 s).

In Figure 40, only the fluid particles are presented and seem to qualitatively follow the expected flow pattern.

The mean velocity values and discharge are computed (Figure 41) and compared at two longitudinal points: x_1 = 2 mm and x_2 = 6 mm (Figure 42).



(**a**) (**b**)

Figure 41. Three-dimensional sudden expansion. Velocity distribution u (y, z) (**a**) x_1 = 2 mm, $\bar{V}_1 = 1.49 \times 10^{-5}\ \mathrm{m/s}, \mathrm{Q1} = (0.4925)^2 \times 10^{-6} \times \bar{V}_1 = 3.61 \times 10^{-12}\ \mathrm{m^3/s}$; and (**b**) x_2 = 6 mm, $\bar{V}_2 = 3.72 \times 10^{-6}\ \mathrm{m/s}, \mathrm{Q}_2 = 3.61 \times 10^{-12}\ \mathrm{m^3/s}$. Relative difference 0%.

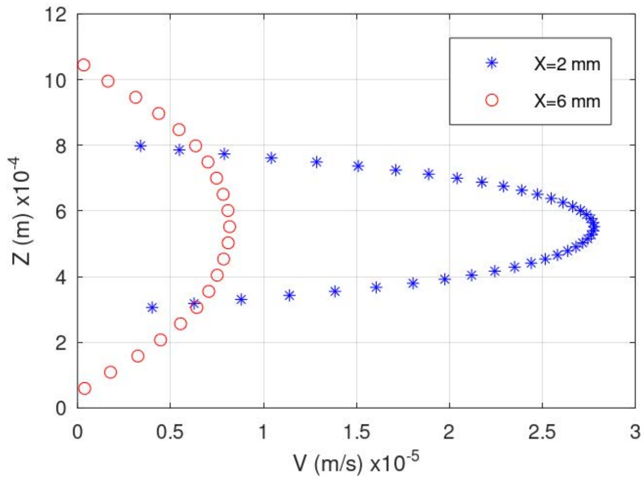


Figure 42. Three-dimensional sudden expansion. Velocity profile u (z, y = 0.4925 mm), x_1 = 2 mm, x_2 = 6 mm.

In the final 3D numerical experiment, we analyze the flow within an extended duct featuring periodic expansions and contractions in its cross-sectional area. Utilizing the microchannel's geometric properties, we optimize the simulation by defining a suitable computational domain (module) where PBCs are applied along the streamwise x-direction. The module has a length of 12 mm, with fluid-occupied cross-sectional areas measuring (0.4925 mm × 0.4925 mm) and (0.985 mm × 0.985 mm). In the simulation, the fluid moves under the action of a body force per unit volume ρg_x, which corresponds to a pressure gradient dp/dx. At time t = 0 the fluid is considered motionless.

Since there is no analytical solution for comparison, examination of the flow field reveals qualitatively reasonable results (Figure 43). To the authors' knowledge, there is no published numerical solution with conventional CFD methods for the problem at hand. Consequently, as a further test, we checked the degree that the SPH solution satisfies the conservation of mass (most importantly to ensure that we do not have leakage of fluid particles through the solid walls). We observed that the continuity equation is confirmed (Figure 44).

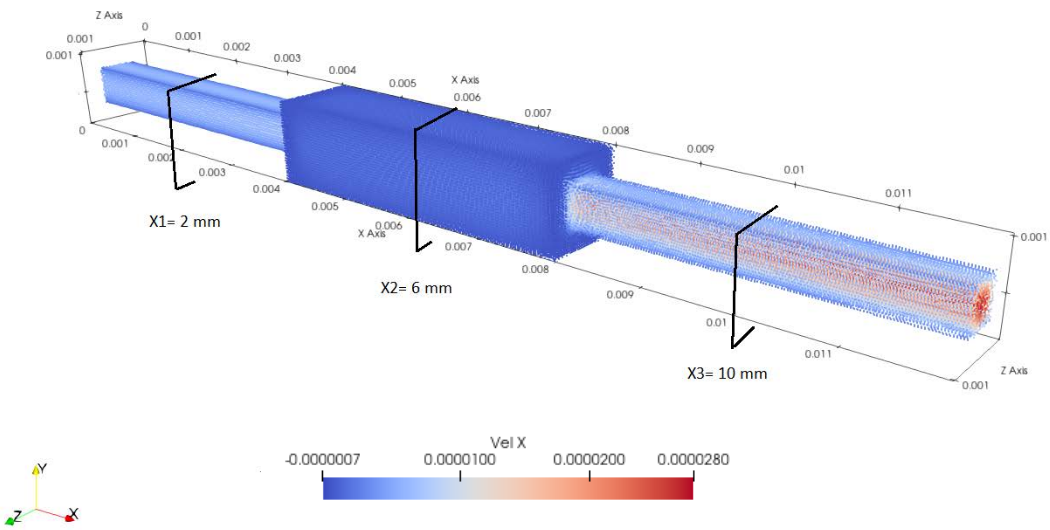
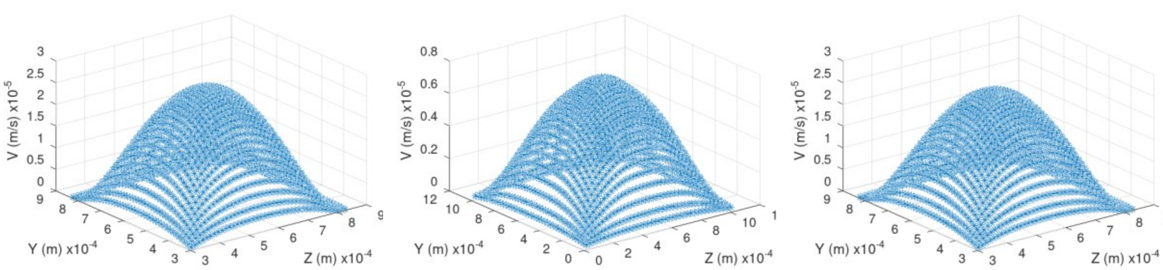


Figure 43. Three-dimensional microchannel with sudden expansions/contractions. The computational domain is a module of the microchannel selected in such a way that periodic BCs can be applied in x-direction. Magnitude of x-velocity component (steady state, t = 50 s).



(**a**) (**b**) (**c**)

Figure 44. Three-dimensional microchannel with sudden expansions/contractions. Velocity distribution u (z, y) for selected longitudinal positions. (a) x_1 = 2 mm, (b) x_2 = 6 mm, and (c) x_3 = 10 mm. $\bar{V}_1 = 1.31 \times 10^{-5}\ m/s, \bar{V}_2 = \ 3.30 \times 10^{-6}\ \mathrm{m/s}$ while $\bar{V}_3 = \ 1.30 \times 10^{-5}\ \mathrm{m/s}$.

As a further check on the accuracy of the SPH method, we examined the linear response of the flow to an increase in the driving force, g_x. The linearity of the response is verified as can be seen in Table 7 with the linear behavior of the volume flow rate (and the velocity, respectively) to be clearly presented in Figure 45.

Table 7. Mean velocity, $\overline{V}$, and volume flowrate, Q, at selected longitudinal stations.

x (mm)	$g_x = 0.001\ m/s^2$		$g_x = 0.002\ m/s^2$	
	$\overline{V}\left(\frac{m}{s}\right) \times 10^{-5}$	$Q\left(\frac{m^3}{s}\right) \times 10^{-11}$	$\overline{V}\left(\frac{m}{s}\right) \times 10^{-5}$	$Q\left(\frac{m^3}{s}\right) \times 10^{-11}$
2	1.31	0.32	2.6	0.63
3	1.31	0.32	2.6	0.63
6	0.33	0.32	0.66	0.64
7	0.33	0.32	0.65	0.63
10	1.30	0.32	2.58	0.63
11	1.30	0.32	2.58	0.63

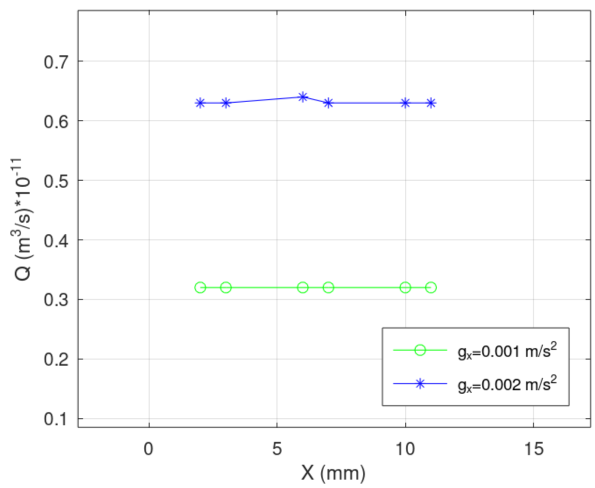


Figure 45. Volume flow rate plot for $g_x = 0.001\ m/s^2$ and $g_x = 0.002\ m/s^2$.

3.3 2D Open Channel-Laminar flow

The simplest standardized test problem involves uniform flow over an inclined plane. The slope is accounted for in the mathematical model through the gravitational acceleration component g_x, which acts in the direction of the flow. To ensure proper boundary conditions, two auxiliary zones are introduced: one at the inlet and one at the outlet. These zones enforce the specified inlet and outlet velocities, as well as the required fluid depth at the inlet (Figure 46). The key parameters for this problem are summarized in Table 8.

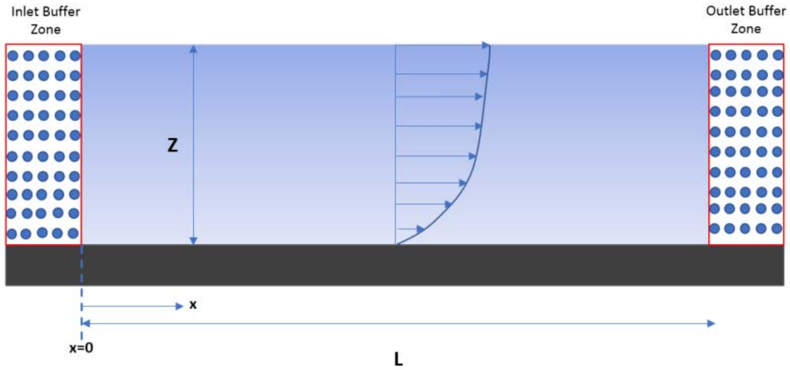


Figure 46. Computational set-up for the 2-D open-channel laminar-flow benchmark: a rectangular domain 8 m long and 1 m deep is flanked by two 0.8 m-long auxiliary zones that impose the prescribed inflow velocity and water depth at the upstream boundary and enforce uniform outflow downstream. A stream-wise gravity component g_x represents the bed slope.

Table 8. Case A: Geometric, Physical, and Numerical Parameters of the Model. N_p represents the number of particles.

L (m)	Z (m)	g_x (m/s^2)	ν_0 (m^2/s)	dp (m)	N_p	T (sec)	Re
8	1	0.855	0.0534	0.08	1757	5	100

The discretization of the medium using particles, along with their initial spatial distribution, is illustrated in Figure 47. After solving the governing equations, the resulting data files are processed using MATLAB packages to generate the velocity profiles shown in Figure 48, while flow snapshots are extracted using Paraview. The analytical solution is derived from (Liakopoulos, 2020).

$$U(y) = \frac{g\ s_0}{2\nu}(2Zy - y^2) \tag{3-7}$$

where s_0 is the channel's slope.

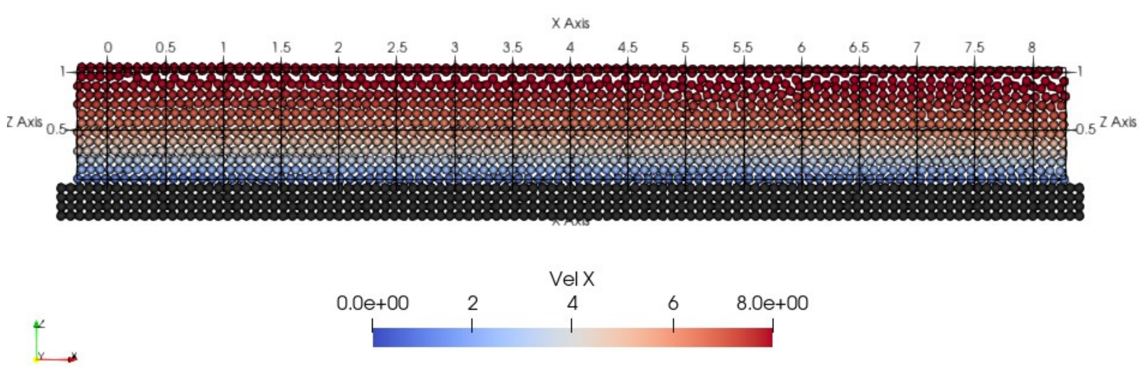


Figure 47. Flow snapshot at t=5 seconds.

As shown in Figure 48, the velocity profile computed using the SPH method is in excellent agreement with the analytical solution, accurately describing a fully developed laminar flow with a Reynolds number of Re=100.

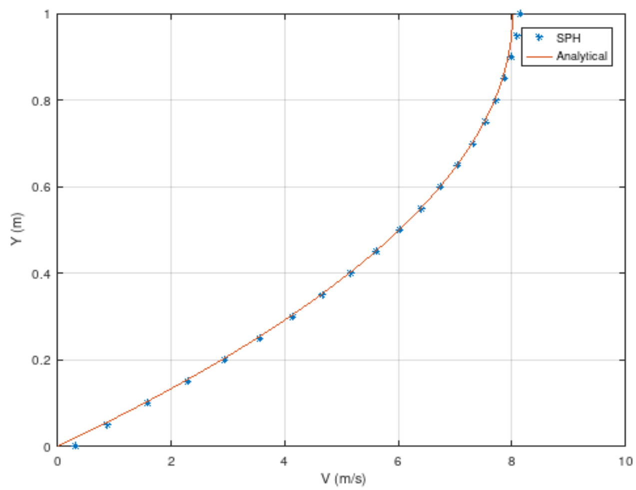


Figure 48. Velocity profile at t=5 seconds at position x=2 m.

3.4 2D Open Channel-Turbulent flow

3.4.1 Hydraulic jump

The main purpose of this chapter (from the study of (Chatzoglou and Liakopoulos, 2022) is to simulate the hydraulic characteristics of several types of 2D hydraulic jumps with upstream Froude number, Fr_1, in the range 2.06 to 4.66. The geometric, physical, and numerical parameters of each simulation are listed in Tables 9 and 10. Simulations ran in parallel CPU mode using the DualSPHysics code (Dominguez et al., 2022).

In the first set of simulations (Figure 49a and Table 9) both upstream and downstream boundary conditions were imposed via buffer zones. A constant-in-time and uniform-in-space velocity was applied as Inlet condition. In addition, a constant surface elevation was assumed at t=0 and hydrostatic pressure was imposed. Outlet conditions were set with a constant-in-time, uniform-in-space velocity, and no other restriction. The value of the outlet velocity is the same as the initial velocity imposed at time t=0 sec for all simulations. Additionally, the water depth is assumed equal to Z_1 at t=0 (Figure 49a) and is free to vary for t>0. Most of the simulations require around 1-2 hours for about 20 sec of simulation time, for N particles (listed in Tables 9 and 10) on a 72-core parallel CPU server.

In order to compare the accuracy, efficiency, and computational cost, some of the cases of Group **I** were remodeled with only difference the outlet condition treatment. In the simulations of Group **II**, the outlet buffer zone was replaced by a weir, an impermeable solid boundary placed perpendicular to the main flow direction (Figure 49b). For the second group of simulations, the height of the weir is not the same in all simulations and is not necessarily equal to the initial flow depth (see Table 10).

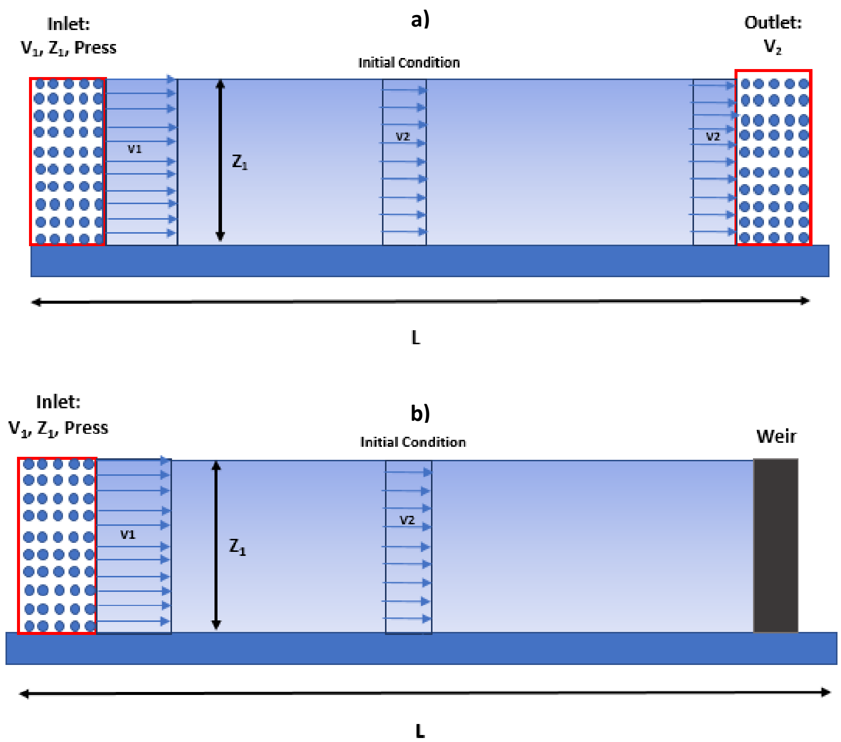


Figure 49. a) Simulation model set up for the In/Out BCs case, b) Simulation model set up for the weir downstream cases. Z_1 refers to the initial water depth.

Table 9. Group **I** model parameters (In/Out BCs with buffer zones).

Case	Fr_1	L (m)	Z_1 (m)	V_{in} (m/s)	V_{out} (m/s)	ν (m^2/s)	Time (sec)	N
Weak jump	2.06	38	0.6	5	2.0	0.00253	20	5287
Oscillating jump	3.30	24	0.6	8	3.6	0.00253	20	3351
Steady jump	4.66	14	0.3	8	2.0	0.00253	40	5335

Table 10. Group **II** model parameters (weir downstream).

Case	Fr_1	L (m)	Z_1 (m)	V_{in} (m/s)	h_{weir} (m)	ν (m^2/s)	Time (sec)	N
Weir A	3.30	38	0.6	8	2.5	0.00253	20	5455
Weir B	3.30	24	0.6	8	2.5	0.00253	20	3530
Weir C	3.30	24	0.6	8	1	0.00253	30	3422
Weir D	3.30	24	0.6	8	1.5	0.00253	30	3458
Weir E	2.06	38	0.6	6	1.5	0.00253	20	5294
Weir F	2.06	24	0.6	6	1.5	0.00253	20	3452

In the initial configuration, the interparticle distance, dp, is set equal to 0.08 m, apart from the steady jump case in which a dp=0.04 m produced better results. For t>0 particles move and any voids that develop due to nonuniform particle displacement are alleviated through the embedded particle shifting technique.

It is well known that for a classical simple hydraulic jump, the ratio of sequent depths (y_1, y_2) is given by

$$\frac{y_2}{y_1} = \frac{1}{2}\left[-1 + \sqrt{1 + 8{Fr_1}^2}\right] \tag{3-8}$$

Moreover, the jump length, normalized by y_2, has been experimentally determined and is presented graphically in standard hydraulics literature (Liakopoulos, 2020; Chaudhry, 2008). In this study, the sequent depths and jump length were obtained by post-processing the SPH results using the data analysis and visualization tool Paraview. A comparison between numerical and theoretical results is provided in Tables 11 and Figure 58 for Group I cases, and in Figure 60 and Table 13 for cases involving a downstream weir (Group II). It is worth noting that specific energy and specific momentum are defined as follows:

$$E_i = y_i + \frac{Q^2}{2g{A_i}^2} \tag{3-9}$$

$$Fs_i = \frac{{y_i}^2}{2} + \frac{q^2}{gy_i} \tag{3-10}$$

In Eqs. (3-9, 3-10), Q and q refer to the discharge and specific discharge respectively while A denotes the cross-section area. The difference between the computed E_1 and E_2 represents energy loss due to both bed friction and the energy dissipation in the jump. The difference between total head upstream and downstream of the jump, h_l, is the theoretical energy loss (within the one-dimensional framework) computed by Eq. 3-11 in terms of the jump's sequent depths

$$h_l = \frac{(y_2 - y_1)^3}{4y_1y_2} \tag{3-11}$$

It should be mentioned that eqs. (3-8 to 3-11) are derived by one-dimensional analysis and neglecting friction at the channel bottom. Total head loss and comparisons between theoretical data and numerical outputs are presented in Table 12.

Weak jump

For Fr_1 between 1 to 1.7 (undular jump) sequent depths y_1, y_2 are almost equal and only a ruffle is formed on the surface. As Fr_1 approaches 1.7, rollers are formed. The energy loss is low (Chaudhry, 2008). As can be seen in Figure 50, after a transient period, the jump reaches an almost steady position, disturbs considerably the free surface and for x ≥ 7m the fluid moves smoothly. The sequent depth y_2 is 1.50 m and the averaged-in-time velocity profiles at several longitudinal positions are shown in Figure 51. Unsurprisingly, in our simulations there is low energy loss in accordance with the weak jump description.

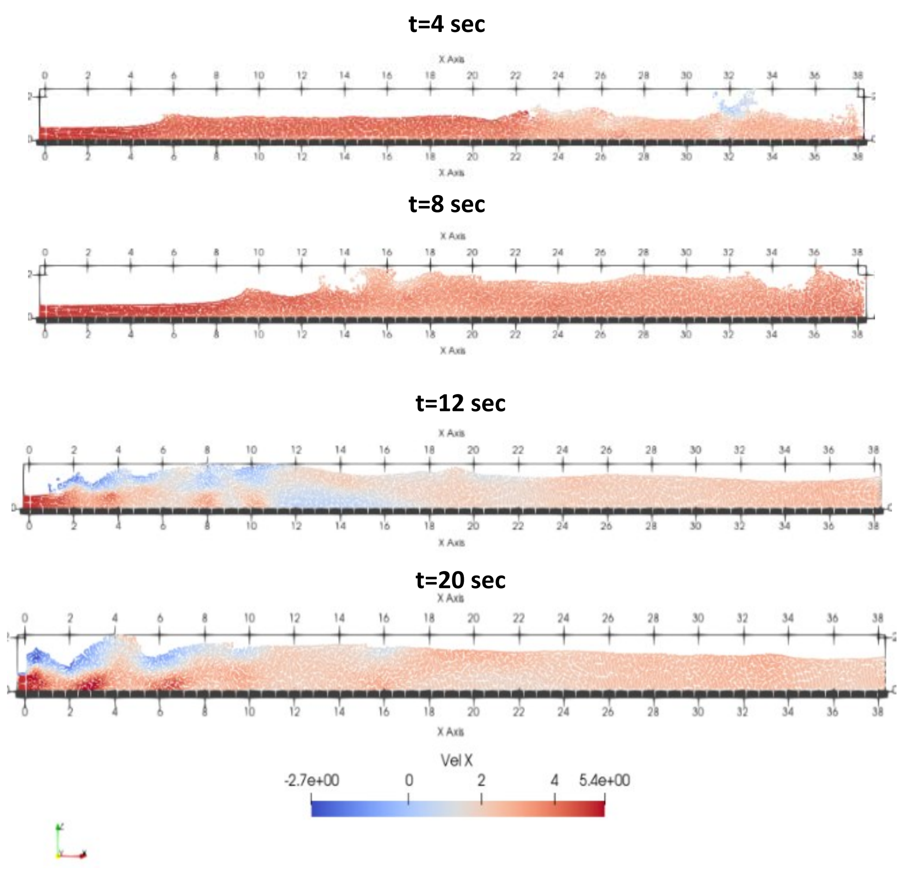


Figure 50. Formation of weak hydraulic jump for Fr_1=2.07. Snapshots of the flow at t=4, 8, 12 and 20 secs.

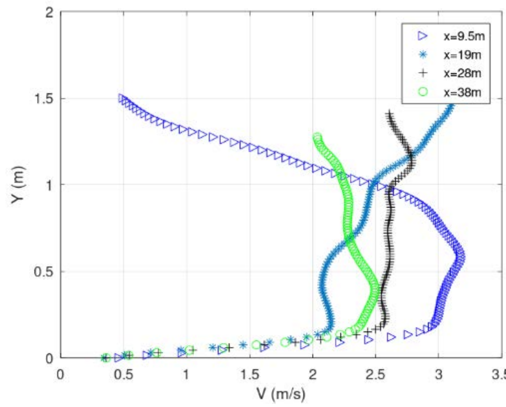


Figure 51. Weak jump. Velocity profiles at various longitudinal positions. V_{mean} (x=9.5 m) = 2.32 m/s, V_{mean}(x=19m) = 2.34 m/s, V_{mean}(x=28m) = 2.49 m/s, V_{mean}(x=38m) = 2.2 m/s.

Oscillating jump

In the case of an oscillating jump, the jet at the jump entrance shifts between the bottom and the surface. Turbulence can occur near the channel bottom at one moment and at the surface the next. These oscillations generate irregular waves, making this range of Fr_1 unsuitable for energy dissipator design (Liakopoulos, 2020). In our SPH simulations for Fr_1=3.3, we observe the development of irregular waves, particularly when the fastest shock wave reaches the outlet. Due to velocity constraints at the outlet boundary, the wave is reflected, interacts with the slower shock wave, and merges with it, reducing its velocity. This interaction results in a nearly stationary hydraulic jump (Figure 52). Additionally, as expected, these waves are accompanied by flow recirculation and free-surface oscillations (Figure 53).

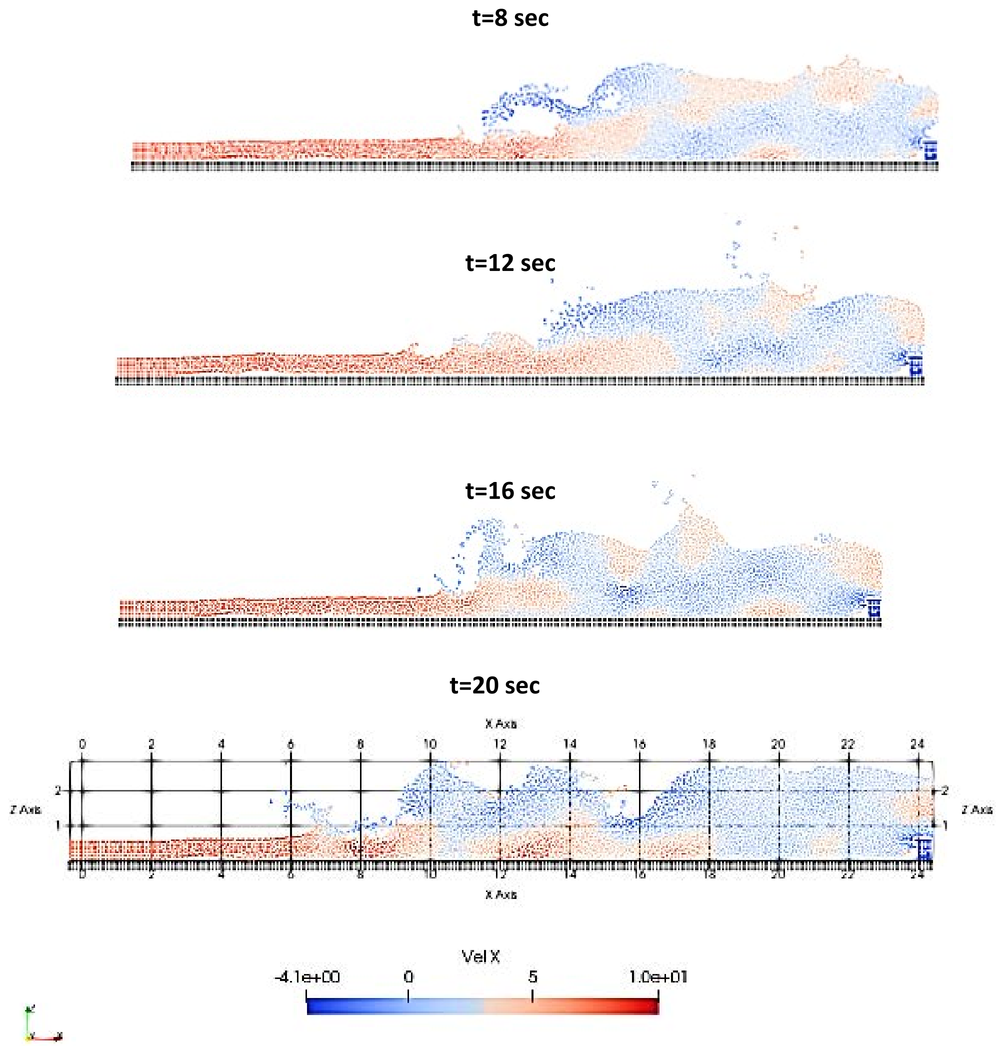


Figure 52. Formation of oscillating hydraulic jump for Fr1=3.30. Snapshots of the flow at t=8,12, 16 and 20 secs.

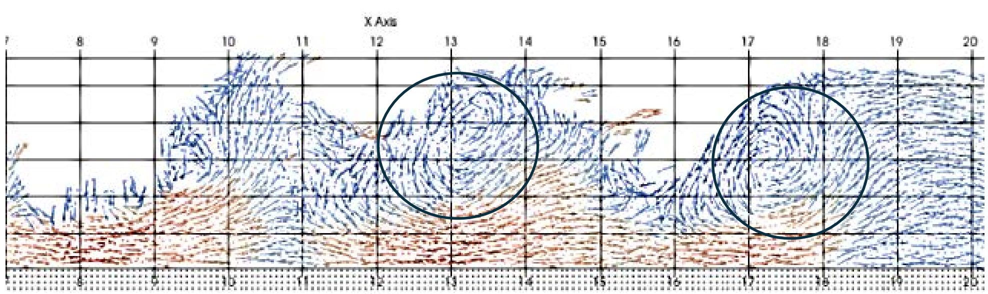


Figure 53. Oscillating jump. Velocity field along the jump.

SPH predicts a high-speed flow for the first 5-6 meters of the channel. The flow velocity is reduced after the jump (Figure 54) as expected. After that, we observe profiles in which the lower part of the velocity exhibits characteristics of approximately regular flow, while the upper part exhibits characteristics due to the highly turbulent structure of the flow. The computed energy loss is higher than in the weak jump case (see Table 12) as expected.

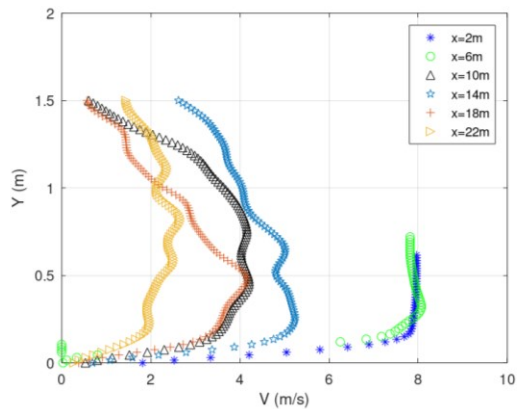


Figure 54. Oscillating jump. Velocity profiles at selected longitudinal positions. V_{mean}(x=2m) = 7.24 m/s, V_{mean}(x=6m) = 6.54 m/s, V_{mean}(x=10m) = 3.22 m/s, V_{mean}(x=14m) = 4.18 m/s, V_{mean}(x=18m) = 2.58 m/s, V_{mean}(x=22m) = 2.04 m/s.

Steady jump

As Chaudhry, (2008) mentions, in the case of a steady jump, the jump is formed at the same location leading to a well-balanced jump with considerable energy dissipation. In our computations, after approximately 20 secs of simulation time, the jump reaches a steady position with high energy dissipation and after a length of almost 9 m the flow reaches a steady state with nearly constant depth y_2 (Figure 55). Flow recirculation is also observed in this case yet has totally vanished for x>9 m (Figure 56). The uniform constant velocity at the Inlet (V_{in}=8m/s) lead to an almost stationary jump in which negative, and zero velocity values are observed. For x greater than 6-7 m, velocity is greatly reduced (Figure 55), having as a result high energy loss, in agreement with laboratory experiments. This type of jump is characterized by high energy dissipation and great velocity reduction. The zero velocity values that appear at x=2 m is a result of the particle nonuniform distribution and poor performance of the particle shifting algorithm which leads to the formation of a gap. Velocity profiles in selected longitudinal locations are depicted in Figure 57.

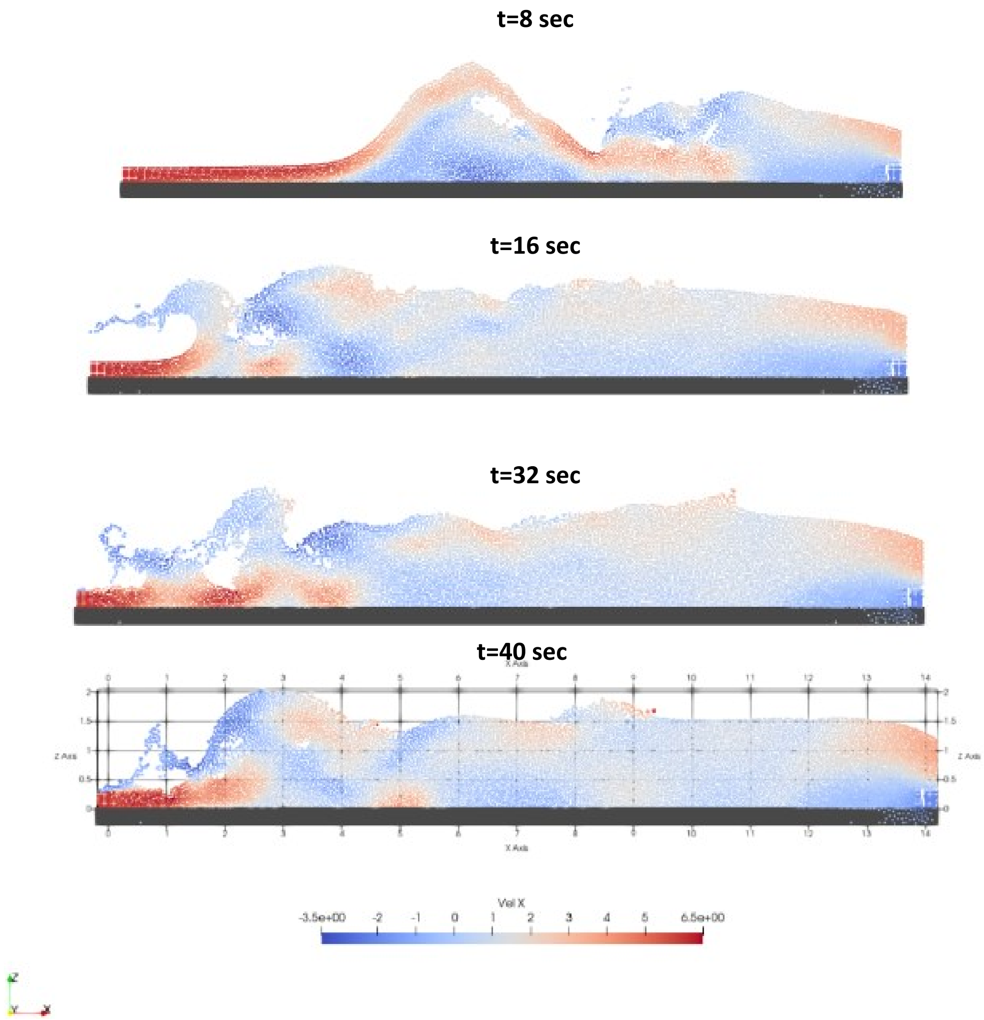


Figure 55. Formation of steady hydraulic jump for Fr_1=4.66. Snapshots of the flow at t=8, 16, 32 and 40 secs.

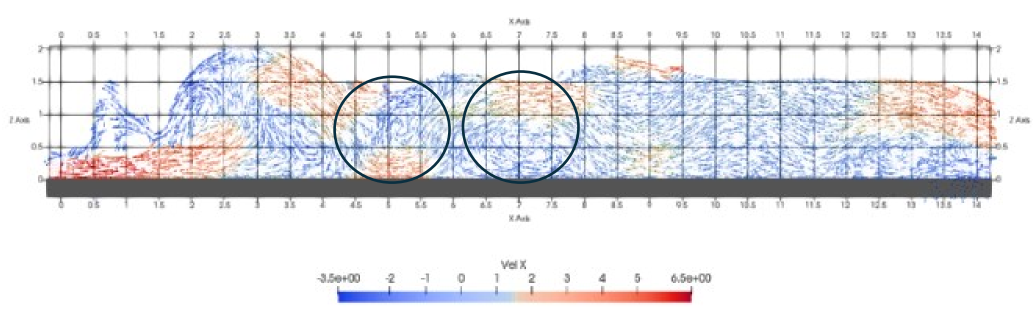


Figure 56. Steady jump. Velocity field at t= 40 secs.

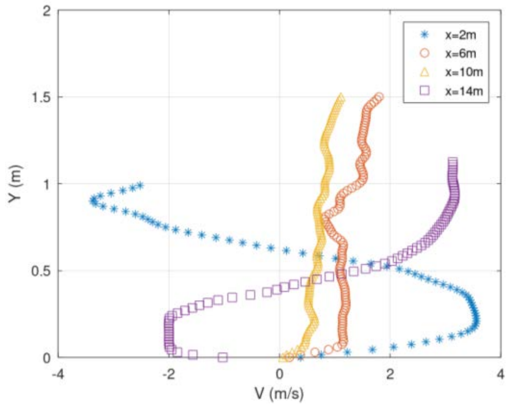
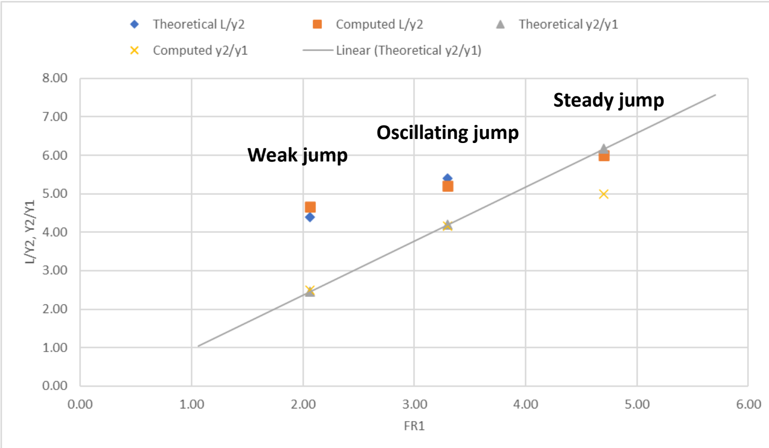


Figure 57. Steady jump. Velocity profiles at various longitudinal positions. Vmean(x=2m) = 0.69 m/s, Vmean(x=6m) = 1.23 m/s, Vmean(x=10m) = 0.71 m/s, Vmean(x=14m) = 1.05 m/s.

Quantitative comparisons

Table 11. Group **I**. Comparison between numerical results and theoretical predictions.

Case	Fr_1	y_2 (comp.)	L (comp.)	Fr_2 (comp.)	L/y_2 (comp.)	y_2/y_1 (comp.)	Fr_2 (theor.)	Fr_2 Diff. %	L/y_2 (theor.)	y_2/y_1 (theor.)
Weak jump	2.06	1.5	7	0.61	4.67	2.50	0.54	10.94	4.40	2.46
Oscil. jump	3.30	2.5	13	0.41	5.20	4.17	0.38	7.90	5.40	4.19
Steady jump	4.66	1.5	9	0.26	6.00	5.00	0.31	16.13	6.00	6.17



Case	Diff. (y_2/y_1) %	Diff. (L/y_2) %
Weak jump	1.60	5.71
Oscillating jump	0.56	3.85
Steady jump	23.40	0.00

Figure 58. Group I. Graphical comparison and relative difference between theoretical predictions and simulation results.

Table 12. Group **I**. Energy loss in the jump. Numerical results, and theoretical predictions.

Case	Energy Loss (m) numerical	Energy Loss (m) theoretical	Difference %
Weak jump	0.17	0.19	10.53
Oscillating jump	1.17	1.16	0.86
Steady jump	1.76	1.68	4.76

Simulations with weir as outlet boundary

In Figure 59 snapshots of the flow are presented for each weir case. In addition, outcomes seem to be similar or slightly more inadequate compared to the cases of In/Out BCs.

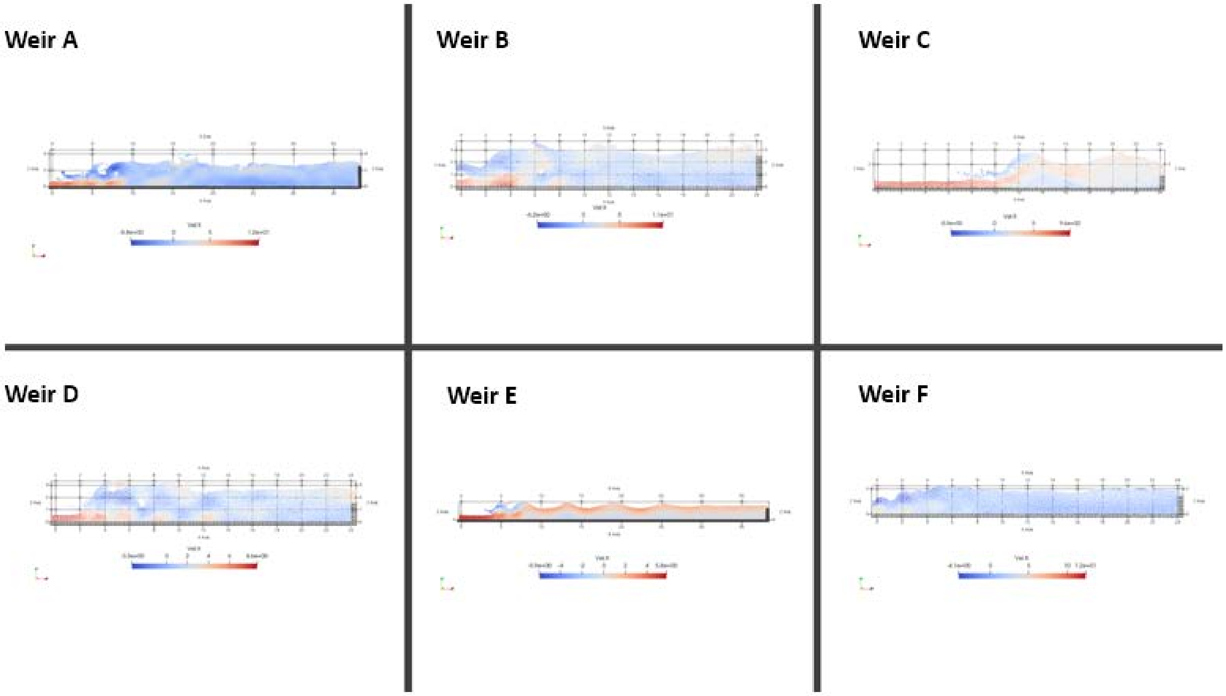
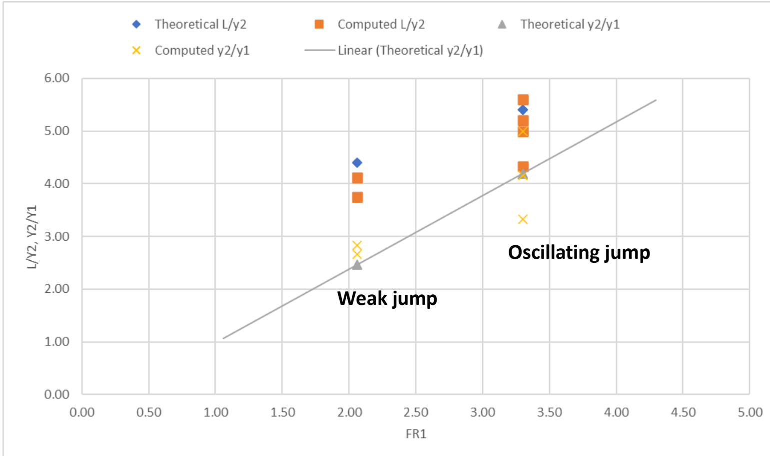


Figure 59. Group II simulations. Formation of hydraulic jump. Flows snapshots corresponding to the cases of Table 10.

Quantitative Comparisons

Table 13. Group **II**. Comparison between numerical results and theoretical predictions.

Case	Fr_1	y_2 (computed)	L (computed)	Fr_2 (computed)	L/y_2 (computed)	y_2/y_1 (computed)	L/y_2 theoretical	y_2/y_1 theoretical
Weir A	3.30	2.5	14	0.18	5.60	4.17	5.40	4.19
Weir B	3.30	3.0	13	0.12	4.33	5.00	5.40	4.19
Weir C	3.30	2.0	10	0.40	5.00	3.33	5.40	4.19
Weir D	3.30	2.5	13	0.13	5.20	4.17	5.40	4.19
Weir E	2.06	1.7	7	0.09	4.12	2.83	4.40	2.46
Weir F	2.06	1.6	6	0.25	3.75	2.67	4.40	2.46



Case	Diff. (y_2/y_1) %	Diff. (L/y_2) %
Weir A	0.56	3.57
Weir B	16.20	24.62
Weir C	25.70	8.00
Weir D	0.56	3.85
Weir E	13.18	6.86
Weir F	7.75	17.33

Figure 60. Group II. Graphical comparison and relative difference between theoretical predictions and simulation results.

Results of the Group **II** cases with the solid boundary (weir) as an outlet confirm that they are more computationally expensive. It is worth mentioning that this computational cost is high, mainly due to the need of additional simulation time to reach an almost steady jump position and not because of the cost of solving the PDEs of the problem.

3.4.2 Sluice Gate-Weir flow control system.

In this group of simulations, we assess the SPH method's performance in turbulent two-dimensional open channel flow. The simulated system consists of two separated water volumes, a sluice gate, and a weir at the outlet. The different zones of fluid are separated with a solid wall (gate) with an opening, Y_G. At time t >0, fluid 1 flows with an imposed initial velocity V_1 while fluid 2 is steady up to the moment that the two zones are unified. To achieve constant inlet conditions, we apply an inlet buffer zone with 5 layers of particles (Figure 61) which has a constant-in-time and uniform-in-space velocity V_1 equal to the initial velocity of fluid 1. The flow depth Z_1 denotes the initial depth of fluid 1, while Z_2 is the initial depth of fluid 2 and is set equal to the gate's opening i.e., $Z_2 = Y_G$. The simulation parameters (geometric, initial conditions and physical properties) are listed in Table 14.

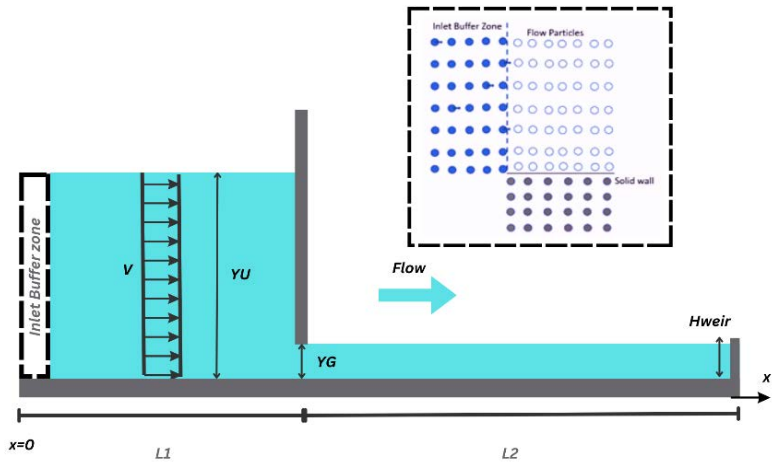


Figure 61. Schematic representation of the computational domain (not to scale). Simulation model set up. L1 and L2 are the lengths of each fluid accordingly. Hweir equals to the height of the weir depending on the case, YU=upstream water depth, while YG is the gate opening.

The first test case, Case A, follows the description of the flow mentioned above. To compare the influence of fluid viscosity and possible the initial conditions two more cases were simulated. Case B differs from Case A in the assigned initial velocity and the fluid viscosity. In addition, to examine how does the placement of the weir affects the hydraulic jump formation and the rest hydraulic characteristics of the flow, Case C, is formed with identical parameters to Case B, but the longitudinal position of the weir. Specific parameters for those cases are listed in Table 14.

Table 14. Model Parameters and initial conditions in each simulation. N is the number of particles.

Case	L_1 (m)	L_2 (m)	Z_1 (m)	α (m)	dp (m)	N	Z_2 (m)	h_{weir} (m)	ν (m^2/s)	V_1 (m/s)	L_w (m)
A	2.5	5.25	0.25	0.04	0.002	234236	0.04	0.05	3.00E-04	0.5	5.24
B	2.5	5.25	0.25	0.04	0.002	234236	0.04	0.05	1.00E-06	0.2	5.24
C	2.5	5.25	0.25	0.04	0.002	234236	0.04	0.05	1.00E-06	0.2	3.24

Simulations run on a parallel CPU server with a capability of 72 cores and for a simulation time of 20 sec, 2-3 days CPU time are needed.

Qualitative analysis

Case A resulted in Submerged flow conditions downstream of the sluice gate. As can be seen in Figure 62, fluid starts flowing under the gate at t>0 and a hydraulic jump starts to form reaching an approximately steady position after 4-5 secs. However, a few seconds later, the downstream tailwater depth is larger than the conjugate depth of the sluice gate opening, forcing the flow to elevate to a depth above the conjugate, leading to submerged (drowned) hydraulic flow conditions (Figure 63).

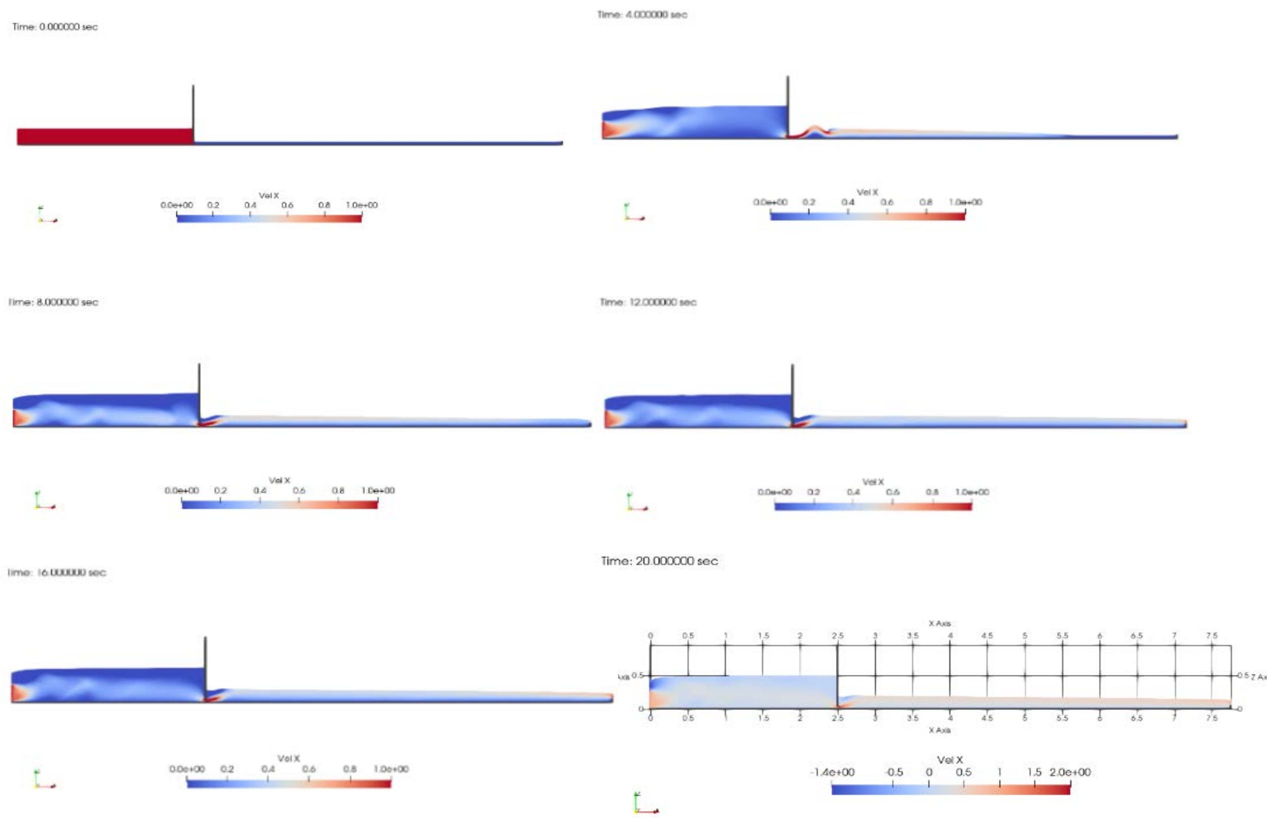


Figure 62. Case A. Flowfield snapshots (from left to right) at t=0 sec, t=4 sec, t=8 sec, t=12 sec, t=16 sec, t=20 sec.

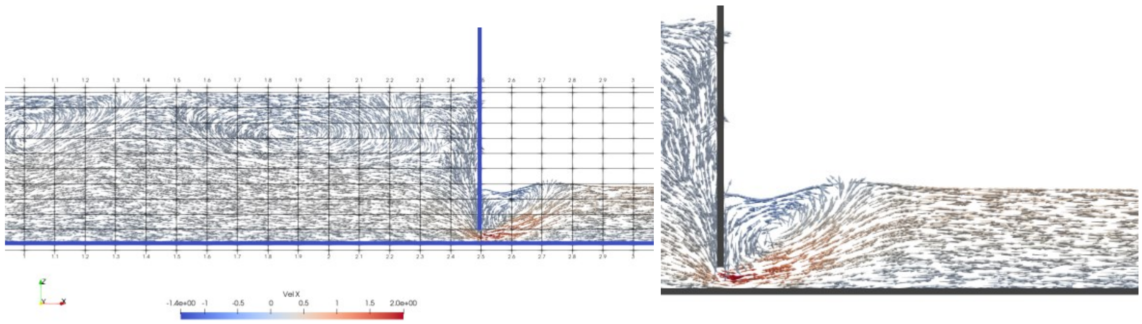


Figure 63. Case A. Velocity field in the vicinity of the sluice gate at t=20 sec. Arrows show the velocity magnitude (color) and the velocity direction of the discretized continuum.

In Case B, free flow conditions prevail immediately after the sluice gate, and the hydraulic jump, formed downstream, reaches a steady position (Figure 64). Some inaccuracies occur due to nonuniform particle distribution and “gaps” result in a rate of uncertainty due to the lack of particles used to solve the PDEs. Though, the model captures the main flow characteristics (Figure 65).

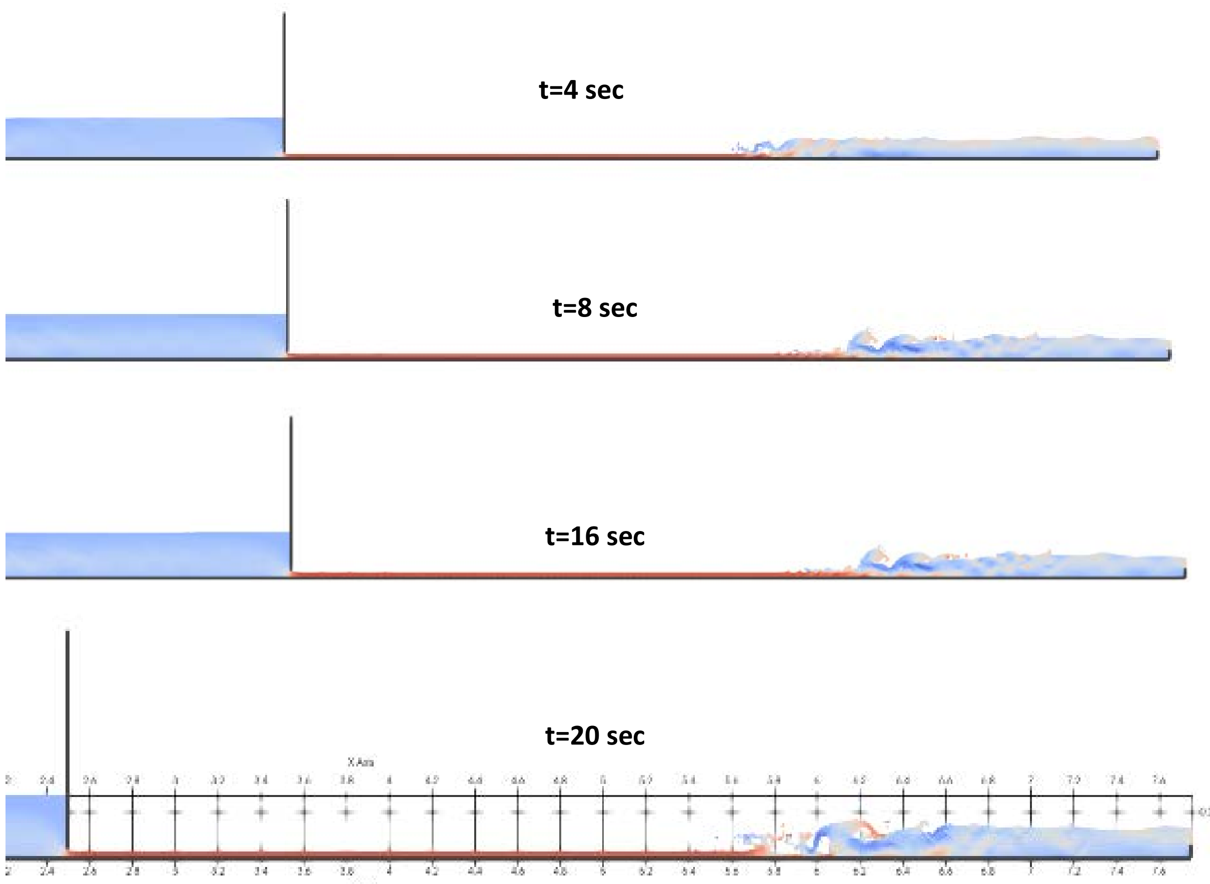


Figure 64. Case B. Weir is placed at x=7.74 m. Flowfield snapshots at (**a**) t=4 sec, (**b**) t=8 sec. (**c**) t=16 sec, (**d**) t=20 sec.

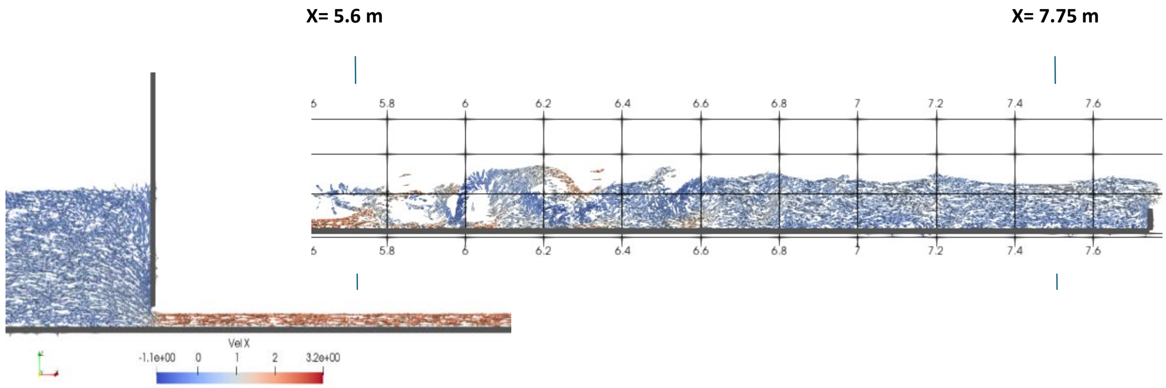


Figure 65. Case B. Velocity field at t=20 sec. Arrows show the velocity magnitude (color) and the velocity direction of the discretized continuum.

Case C, similar to Case B, resulted in free water flow downstream the sluice gate. Placement of the weir in a position closer to the gate gave a faster formation and an almost steady position of the jump (at 6-7 secs) as it is shown in Figure 66. There are still some inaccuracies due to the particle distribution and the nature of the hydraulic jump. Yet, the simulation captures the main flow characteristics (see Figure 67).

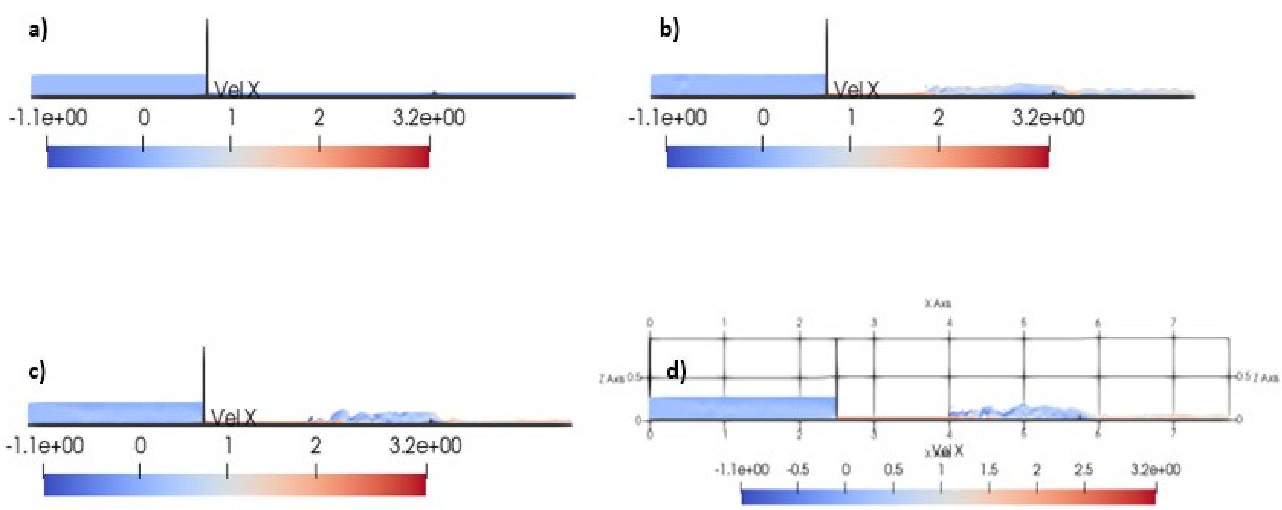


Figure 66. Case C. Weir placed at x=5.74 m. Flowfield snapshots at (**a**) t=0 sec, (**b**) t=8 sec. (**c**) t=16 sec, (**d**) t=20 sec.

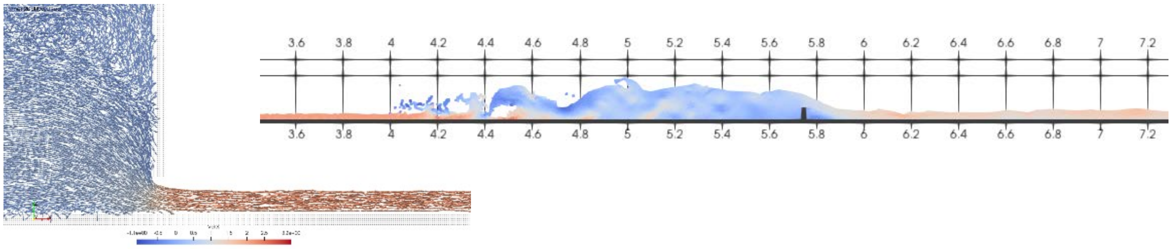


Figure 67. Case C. Velocity field at t=20 sec. Arrows show the velocity magnitude (color) and the velocity direction of the discretized continuum.

Quantitative analysis

In order to have a clearer quantitative comparison for the three cases we separate the analysis into the influence of the sluice gate and the formation of the hydraulic jump.

Sluice Gate

Henry's (1950) experimental study on the discharge coefficient for flow under a sluice gate is considered reliable for both free and submerged flow conditions. Henry presented his laboratory measurements graphically in the form $C_d = f(\frac{h_0}{\alpha}, \frac{h_2}{\alpha})$, where C_d is the discharge coefficient, α is the sluice gate opening, h_0 the upstream water depth and h_2 the tailwater depth. Swamee (1992) has reanalyzed Henry's graphs and proposed algebraic expressions for the sluice gate discharge equations. We briefly describe Swamee's formulae to facilitate comparisons with our SPH results. Let

$$h_{0max} = 0.81 h_2 \left(\frac{h_2}{\alpha}\right)^{0.72} \tag{3-12}$$

Then, for $h_0 \geq h_{0max}$ we have free flow conditions under the sluice gate and the specific discharge (i.e., flowrate per unit length of the sluice gate) is given by

$$q = \frac{Q}{b} = 0.864 a \sqrt{g h_0} \left(\frac{h_o - \alpha}{h_0 + 15\alpha}\right)^{0.072} \tag{3-13}$$

When , $h_2 \leq h_o < h_{0max}$ we have submerged flow conditions, and the specific discharge is given by

$$q = 0.864 \alpha \sqrt{g h_0} \left(\frac{h_0 - \alpha}{h_0 + 15a}\right)^{0.072} (h_0 - h_2)^{0.7} \left\{0.32 \left[0.81 h_2 (\frac{h_2}{\alpha})^{0.72} - h_0\right]^{0.7} (h_0 - h_2)^{0.7}\right\}^{-1} \tag{3-14}$$

Turning now to our SPH results, we observe the following. In Case A, the inequalities $h_2 < h_0 < h_{0max}$ are valid. Swamee's empirical relations predict the formation of submerged flow immediately after the sluice gate. This is indeed the character of the flow computed by SPH (Figures 62 & 63). In contradistinction, Cases B and C are characterized by free flow conditions immediately after the sluice gate, as is shown in Figures 65,67. The condition $h_0 \geq h_{0max}$ (see Table 15) is satisfied. It follows that our SPH results agree with the Henry-Swamee analysis. In Table 15 we summarize the numerical results and empirical predictions (computed by Eqs. 3-12 to 3-14) of the specific discharge (q), upstream depth (h_0), tail-water depth (h_2) and the maximum value of h_0 (h_{0max}).

Table 15. Comparison of SPH numerical results with Swamee's sluicegate predictions.

Case	**h_0 (m)**	**h_{0max} (m)**	**h_2 (m)**	**Flow conditions**	**q (SPH)**	**q (Swamee)**
A	0.250	0.516	0.200	Sub/ed	0.047	0.024
B	0.250	0.246	0.130	Free	0.062	0.049
C	0.250	0.230	0.125	Free	0.061	0.049

<u>Hydraulic Jump.</u>

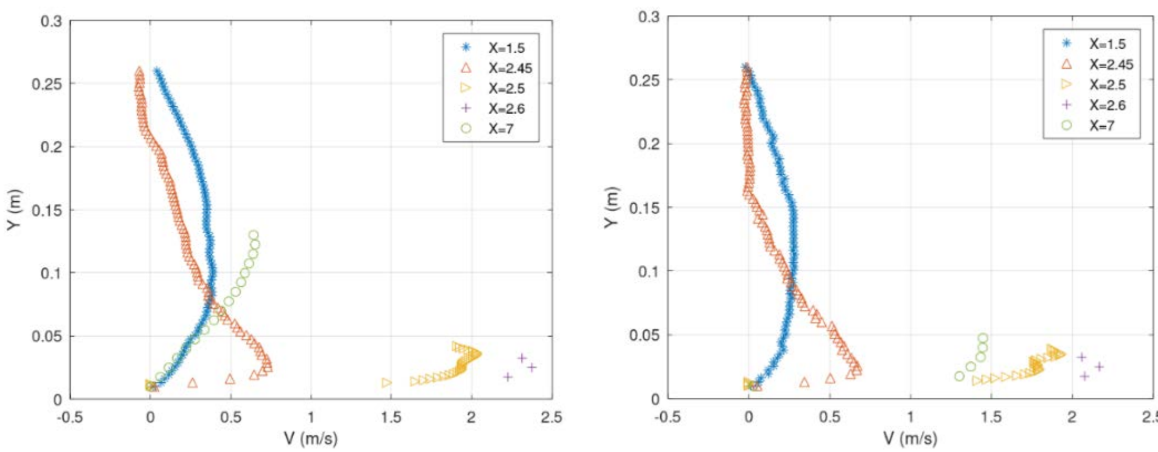
In Case A it is difficult to distinguish any characteristics of the hydraulic jump due to the presence of submerged flow conditions. However, in Cases B,C the free flow conditions give the chance for a better observation of the hydraulic jump formations. In both cases, we have a jump characterized by the formation of rollers on the water surface and the formation of irregular waves which persist for a distance downstream of the jump. Comparing them with Chaudhry's (2008) evaluation of distinct flow patterns for several forms, we conclude that both cases belong to the oscillating jump type.

Further numerical results are compared with the one-dimensional analysis predictions and presented in Table 16. The upstream Fr_1 number in both cases equals to 2.54, which again is a means to classify the jump as oscillating (Liakopoulos, 2020)

Table 16. Comparison of SPH results with experimental-empirical estimations. L_j denotes the hydraulic jump length.

Case	***y_1* *(SPH)***	***Fr_1* *(SPH)***	***y_2* *(SPH)***	***L_j* *(SPH)***	***y_2* *(theoret.)***	***L_j* *(experim.)***	***Diff. in L_j (%)***	***Diff. in y_2 (%)***
B	0.04	2.54	0.130	0.65	0.125	0.61	6.1	3.8
C	0.04	2.54	0.125	0.70	0.125	0.61	12.8	0

Velocity profiles at selected longitudinal locations are depicted in Figure 68a for Case B and Figure 68b for Case C accordingly. The choice of the authors to present velocity profiles only for the last two cases is relevant with the free flow conditions that prevail in those cases and thus from their velocity profiles more notable results can be excluded.



(**a**) (**b**)

Figure 68. Velocity profiles at selected longitudinal locations. The gate is located at x=2.5 m. a) Case B, b) Case C.

3.5 3D Open Channel-Turbulent flow

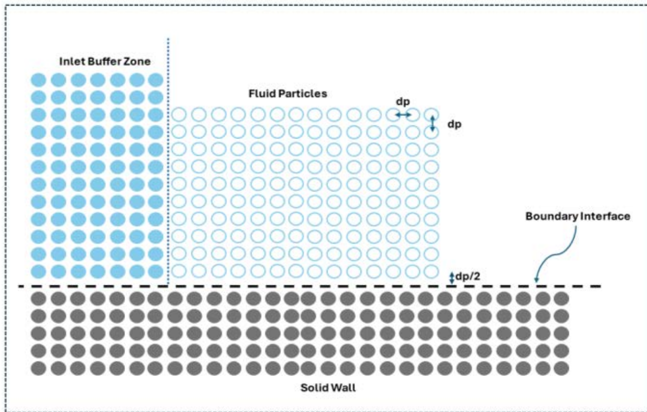
3.5.1 Sluice Gate- Weir flow control system

To assess the effectiveness of the SPH method in the simulation of rapidly varying flows such as the flow under a sluice gate and the potential formation of a hydraulic jump 3D numerical experiments were contacted. Results are validated against experimental data and theoretical solutions.

The simulation model parameters are selected to align with experimental data from YooseffDoost et al. (2022) for comparison. The initial upstream water depth is set to 20 cm, with a channel that is 6.14 m long and 15 cm wide. A sluice gate is positioned at x = 2.02 m, while a weir is placed at the outlet. Further details on the simulation model's geometric setup are illustrated in Figure 69. All simulations are executed on a GPU using an RTX 3080 graphics card.

Primary emphasis is given to the treatment of boundary conditions. Inlet and outlet boundary conditions are enforced along the streamwise direction, while mDBCs are utilized for simulating solid boundaries. Within the inlet buffer zone, both water depth and velocity are maintained constantly to ensure a steady volume flow rate throughout the channel. Given the small gate opening length of YG = 0.025 m, the initial interparticle distance is selected to allow at least five particle layers to pass through the sluice gate.

Consequently, dp = 0.005 m is chosen, with an optimal smoothing length ratio of h/dp = 1.5.



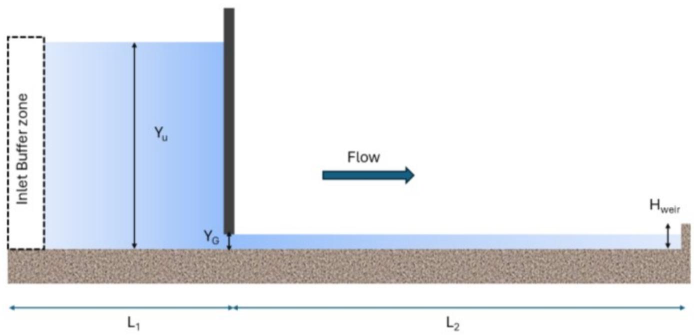


Figure 69. Simulation model set up. L1=2.02 m, L2=4.12 Hweir=height of the weir depending on the case, YU=upstream water depth (initial)=20 cm, while YG is the gate opening.

Several test cases with the same geometry have been tested. In all cases the normal vector which is a key parameter for the appropriate implementation of the mDBCs is evaluated. One important parameter is the so called “ViscoBound” factor parameter. The “ViscoBound” (VB) parameter relates the viscosity of the fluid with the viscosity of the fluid at the liquid/solid interface. When the artificial viscosity scheme is used the parameter α tunes the viscosity of the fluid while α_{BF} the viscosity at the liquid/solid interface. $\alpha_{BF}=\alpha\times VB$. When the Laminar+SPS viscosity treatment is selected the user sets the kinematic viscosity values ($\nu_0=10^{-6}$ m^2/s in our cases). VB serves under the same properties. In this paper the best simulations are presented related to the combination of VB factor and precise application of the mDBCs. Important simulation parameters of each case can be found in Table 17.

Table 17. 3D Cases. Key simulation parameters.

Case	Y_G (m)	Y_U (m)	H_{weir} (m)	N_p	Viscosity treatment	Simulation time (s)
AVM	0.025	0.20	0.025		Artificial	200
L-SPS	0.025	0.20	0.025		Laminar+SPS	200

Case AVM, artificial viscosity treatment.

In the first case (AVM), three different VB factors, VB 1, 10, and 100 were tested. The differences between them were related mainly to the position of the formation of the hydraulic jump and the formation of coherent flow structures, mainly upstream the sluice gate. This study presents the most effective result among the three attempts (VB=100). The jump forms immediately after the gate at a position of x= 2.06 m. Figure 70 provides further details, including a color-coded representation of the velocity-x component.

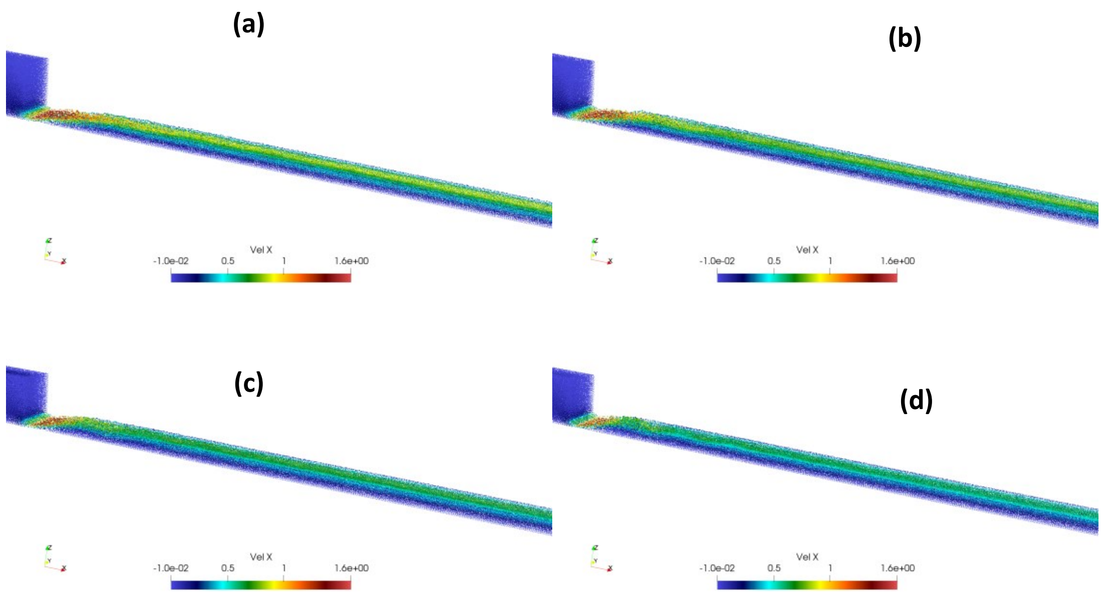


Figure 70. AVM**.** Formation of the Hydraulic jump. a) T=50 sec, b) T= 100 sec, c) T= 150 sec, d) T= 200 sec.

Except for the flow under the sluice gate and the formation of the hydraulic jump it is important to assess SPH method's performance in capturing coherent flow structures such as vortices. Roth and Hager, (1999) from experimental observations, presented a definition sketch of the standard gate with edge vortices. A resemble of their findings is presented in Figure 71.

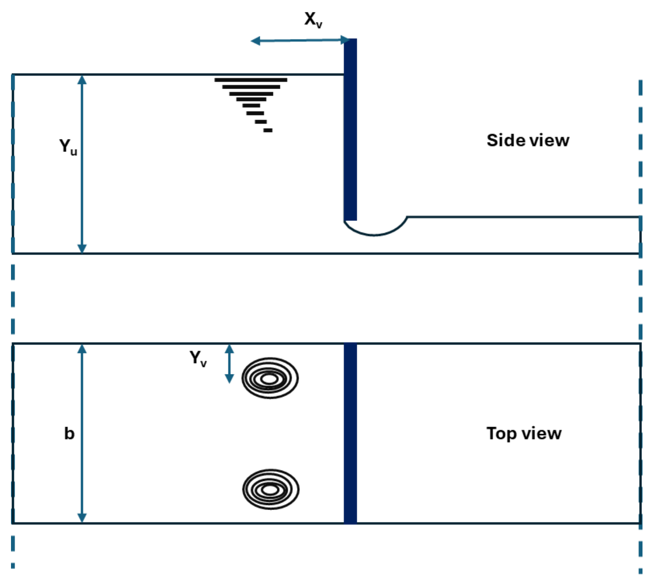


Figure 71. Definition sketch for corner cortices upstream of a Sluice gate.

The formation of these vortices was distinctly captured in our SPH simulation. The temporal development of vortex formation near the water free surface (at Z=18 cm) is illustrated in Figure 72.

The SPH model effectively reproduced the coherent flow structures described in (Roth and Hager, 1999). However, to ensure the accuracy of these results, it was crucial to compare them with the experimental data obtained by Roth and Hager. Their study involved a series of experiments conducted in a channel with a width of 500 mm, 350 mm, and 245 mm, and a total length of 7 m. From these experiments, they derived specific equations to determine the vortex formation locations along the X-axis (streamwise) and Y-axis (spanwise), with the expressions for x_v and y_v presented as follows:

$$x_v = \frac{1}{2}A^{-0.75}, \qquad y_v = \frac{2}{3}\exp\left(-3Y_G\right) \tag{3-15}$$

where A is the relative gate opening (Y_G/Y_U) and $x_v = \frac{X_v}{Y_G}, y_v = \frac{Y_v}{Y_G}$.

For the present hydraulic system, Eq. 3-15 predicted vortex center coordinates of X_v=0.06 m and Y_v=0.016 m. However, the SPH simulation results indicated that the vortex centers were located at X_{v-SPH}=0.08 m and Y_{v-SPH}=0.04 m. It is important to note that the channel width used in our simulations was narrower compared to the experimental setups. According to (Rothe and Hager, 1999), vortex formation tends to occur closer to the wall in wider channels than in narrower ones, which may explain the observed difference.

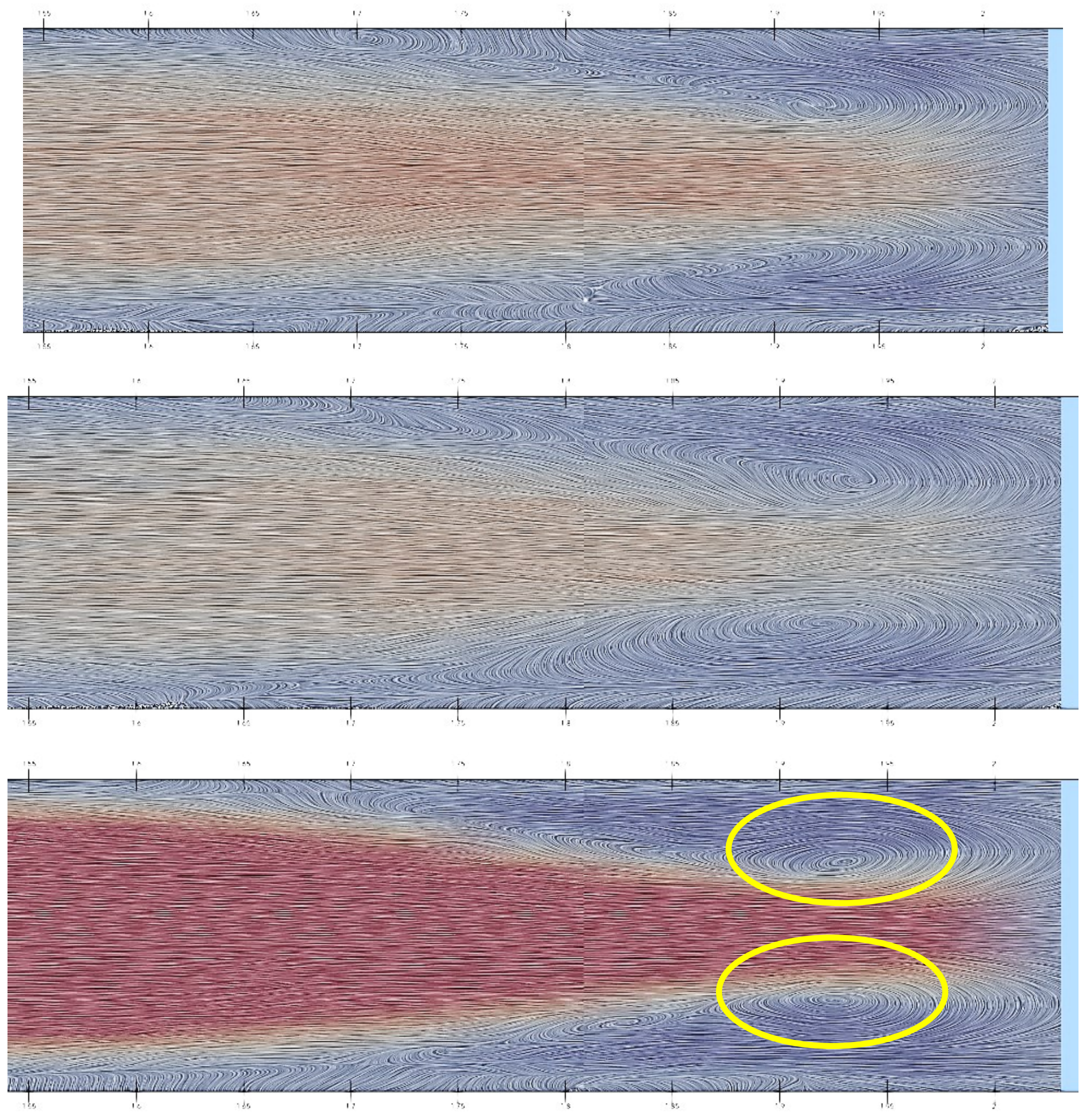


Figure 72. AVM. The transient formation of the coherent vortices upstream of the sluice gate, as seen on a horizontal plane Z= 18 cm. T= 100 sec, T=150 sec, and T= 200 sec.

The flow is in a transient state for about 120 s. Time series plots correspond at points upstream of the sluice gate: point (X = 1.81 m, Y = 0.075 m, Z = 0.005 m), point (X = 1.81 m, Y = 0.075 m, Z = 0.14 m), and (X = 1.81 m, Y = 0.075 m, Z = 0.18 m); see Figure C1 (Appendix C). They correspond downstream from the sluice gate and beyond the influence of hydraulic jump at point (X = 3 m, Y = 0.075 m, Z = 0.005 m), point (X = 3 m, Y = 0.075 m, Z = 0.035), and (B3, X = 3 m, Y = 0.075 m, Z = 0.05 m); see Figure C2 in Appendix C. Figures 73 and 74 illustrate color-coded velocity distribution at cross-sections X = 1.81 m and X = 3m, respectively. The velocity distribution is as expected qualitatively in both cross-sections.

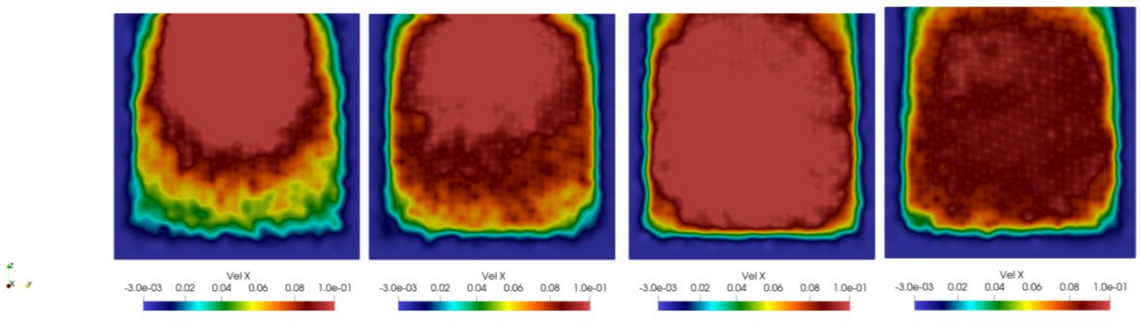


Figure 73. AVM. X-component of velocity. Front view. X= 1.81m, T= 100, T= 150, T= 200 sec.

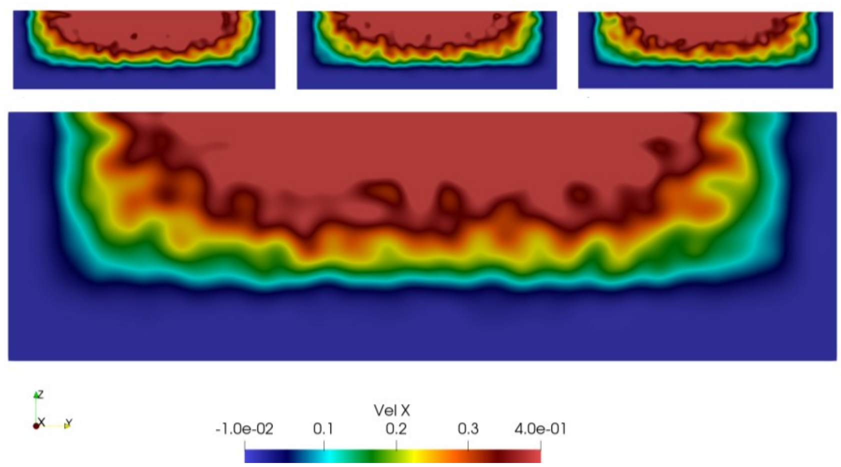
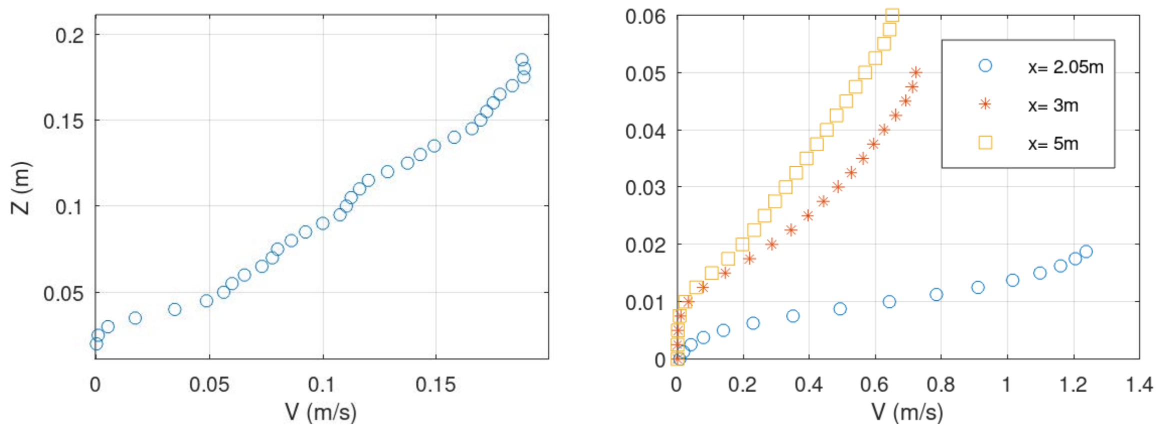


Figure 74. AVM. X-component of velocity. Front view. X= 3m, T= 50, T= 100, T= 150, T= 200 sec.

Mass conservation was achieved as anticipated in a well-executed SPH simulation. It is important to note that, provided fluid particles are prevented from leaving the computational domain through the channel's solid walls, either due to proper parameter selection or adequately designed solid boundaries, the mass conservation principle is inherently maintained (Q_{inlet}=2.5 L/s). This is attributed to the constant number of fluid particles throughout the simulations. The volume flow rate and mean velocity at each location are summarized in Table 18. Velocity profiles at y= 0.75 m at selected longitudinal positions can be found in Figure 75.

Table 18. AVM. The mean velocity and water depth at selected at selected cross-sections.

V_{1_81}=0.093 m/s	V_{2_05}=0.9 m/s	V_3=0.36 m/s
Z_{1_81}=0.18 m	Z_{2_05}=0.0187 m	Z_3=0.05 m
q_{1_81}=0.0168 m^2/s	q_{2_05}=0.0168 m^2/s	q_3=0.0168 m^2/s



(a) (b)

Figure 75. AVM. Velocity profile at selected longitudinal positions. Y= 0.75 m half width of the channel. a) Position upstream of the Sluice gate, X= 1.81 m. b) Positions downstream of the Sluice gate.

The hydraulic jump is formed a few centimeters after the sluice gate. A series of small rollers develop on the surface, yet the downstream water surface remains relatively smooth (Figure 76). These characteristics classify the jump as a weak jump type (Liakopoulos, 2019). Yet, some oscillations occur after the jump, which is not necessary in line with the description of a weak jump type. The Froude number of the jump equals to 2.1 verifying that it is classified as a weak jump according to its Froude number (Fr_1=2.1) (Liakopoulos, 2019).

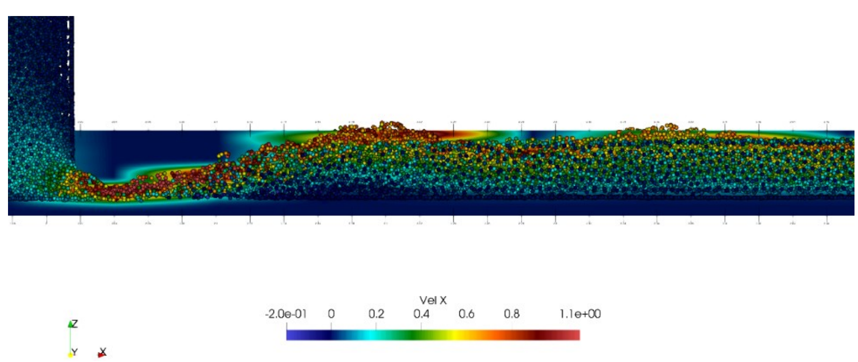


Figure 76. Side view Y=0.75 m. Characteristics of the hydraulic jump at T= 200 sec.

Table 19 presents a comparison between the simulation results and the standard 1D analysis of the hydraulic jump. Although the relative difference of 6.38% is not ideal, it is important to note that this is a comparison between a 3D case and the results from a 1D analysis.

Table 19. Comparison of SPH results with experimental-empirical estimations.

AVM	y_2	SPH (3D)	Theoretical (1D)	Rel. diff %
		0.05	0.047	6.38 %

Case L-SPS, Laminar+SPS viscosity treatment.

Keeping the same geometry and parameters but with a Laminar+SPS viscosity treatment instead, Case L-SPS tested for VB=1, 10 and 100. Again, the VB=100 produced the best results. In that case the hydraulic jump forms at X= 2.25 m (see Figure 77).

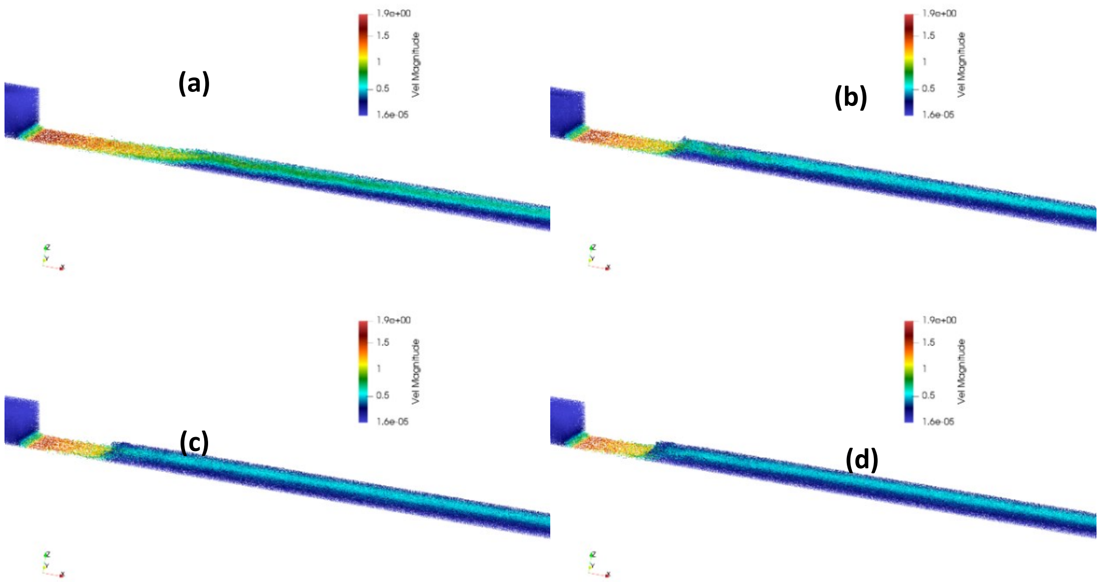


Figure 77. L-SPS. Formation of the Hydraulic jump. a) T=50 sec, b) T= 100 sec, c) T= 150 sec, d) T= 200 sec.

The expected vortices upstream of the sluice gate were successfully generated in this case, appearing in a more distinct and organized form compared to previous observations. According to Eq. 3-15, the vortex center coordinates are calculated as X_v=0.06 m and Y_v=0.016 m. In the L-SPS simulation results, the vortex centers were identified at X_{v-SPH}=0.06 m = 0.06 , (matching the predicted value) and Y_{v-SPH}=0.04 m, showing a significant deviation from the theoretical calculation. This difference is likely caused by the channel's restricted width (Roth and Hager, 1999).

Additionally, the simulation revealed the presence of secondary vortices within the flow pattern (see Figure 78). This feature could play a significant role in the design and analysis of such systems. For instance, secondary vortices contribute to increased turbulence, particularly in high-velocity flow conditions. Moreover, they can impact sediment transport by altering deposition and scouring processes, which are essential factors for ensuring the system’s durability and effective environmental management.

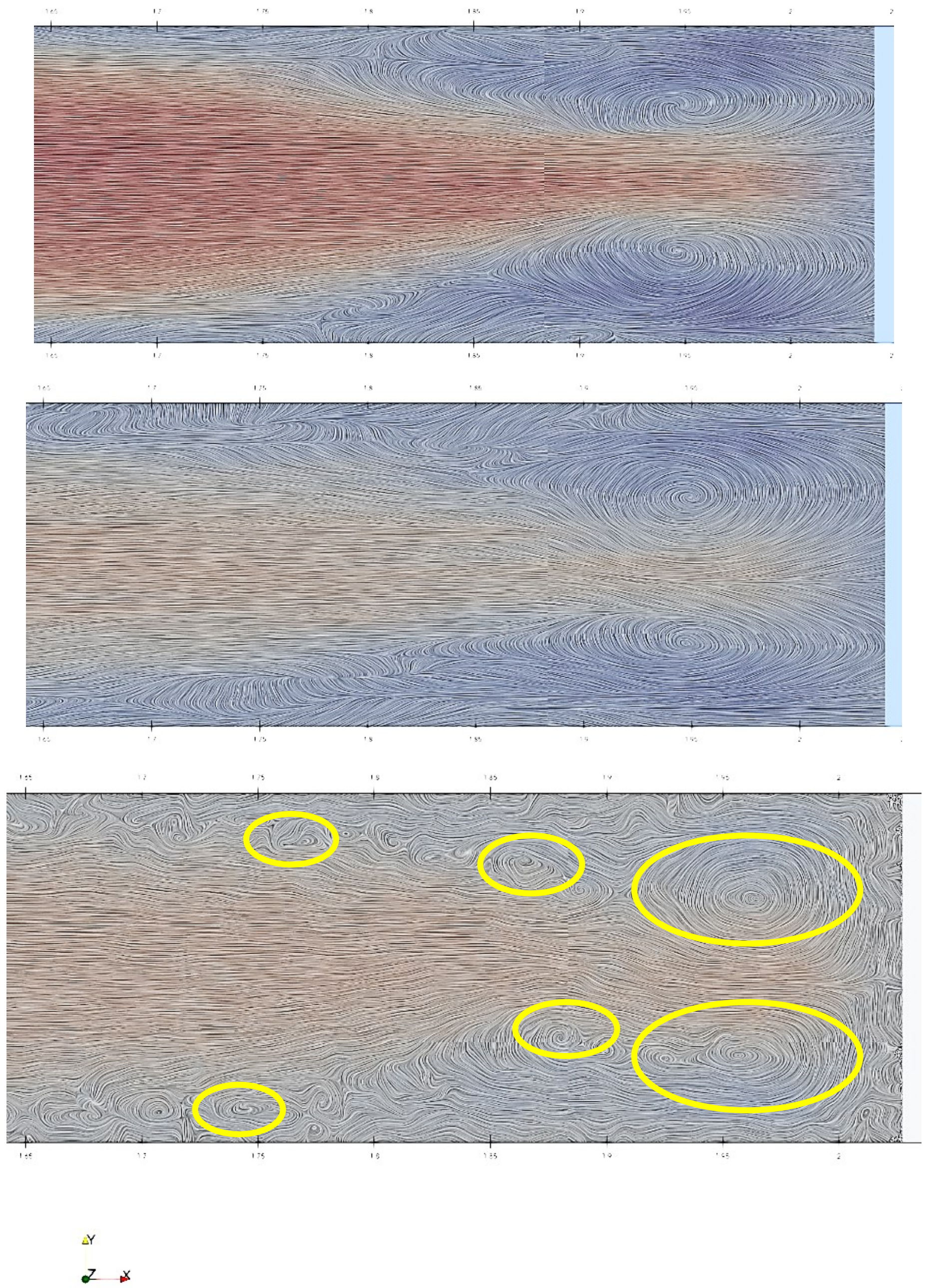


Figure 78. L-SPS. The transient formation of coherent vortices upstream of the sluice gate as seen on a horizontal plane at Top view Z=0.18 m, T= 100 sec, T=150 sec, T= 200 sec.

The flow remained in a transient state for approximately 140 seconds. Time series plots were generated for locations upstream of the sluice gate, specifically at points A1, A2, and A3 (refer to Figure C3). Additionally, plots were obtained for points located downstream of the sluice gate and beyond the hydraulic jump's influence, namely B1, B2, and B3 (see Figure C4). Figures 79 and 80 illustrate the velocity distribution at the cross-sections X=1.81 m and X=3 m using color-coded representations. The velocity patterns at both cross-sections qualitatively matched the expected flow behavior.

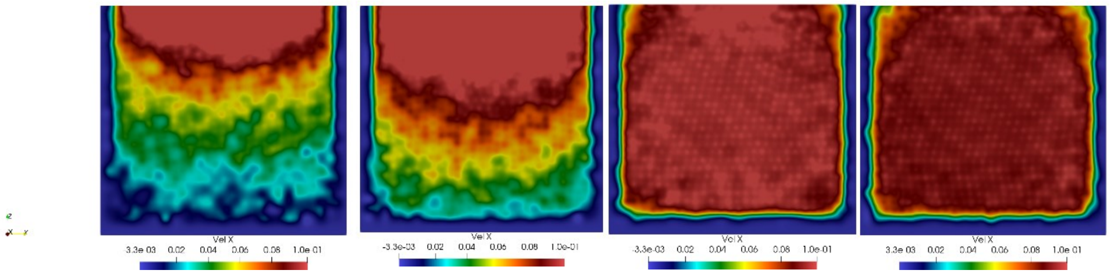


Figure 79. L-SPS. X-component of velocity. Front view, X= 1.81 m, T= 50, T= 100, T= 150, T= 200 sec.

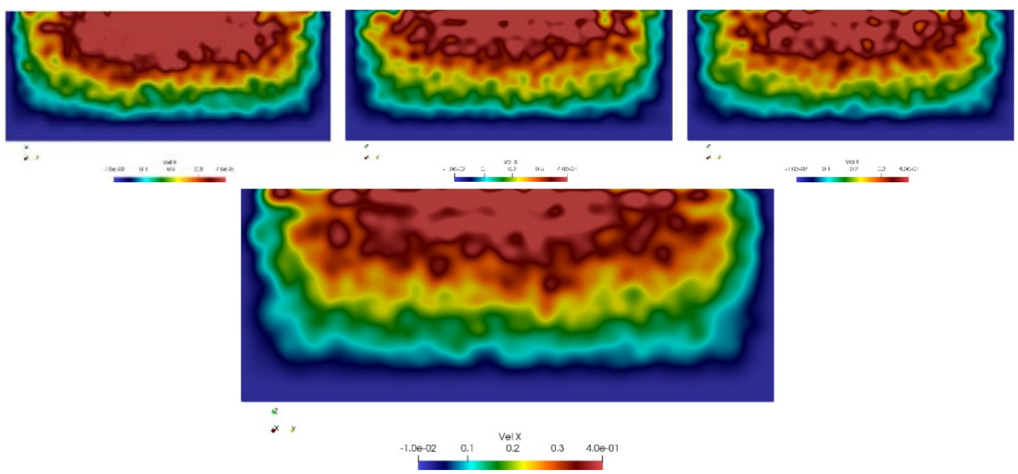
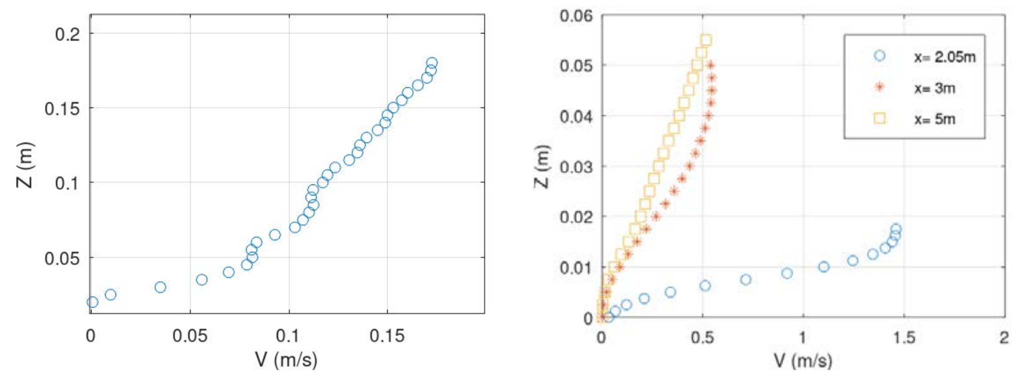


Figure 80. L-SPS. X-component of velocity Front view. x= 3m, T= 50, T= 100, T= 150, T= 200 sec.

Velocity profiles at y= 0.75 m at selected longitudinal positions can be found in Figure 81. The volume flow rate and the mean velocity value for each location are presented in Table 20. The mass conservation is satisfied as expected. (Q_{inlet}=2.5 L/s).



(**a**) (**b**)

Figure 81. L-SPS. The velocity profile at selected longitudinal positions. Y= 0.75 m half width of the channel. a) Position upstream of the Sluice gate, X= 1.81 m. b) Positions downstream of the Sluice gate.

Table 20. Case 2. Mean velocity value and water depth at selected points.

Case 2	V_{1_81}=0.093 m/s Z_{1_81}=0.18 m q_{1_81}=0.0168 m^2/s	V_{2_05}=0.96 m/s Z_{2_05}=0.0175 m q_{2_05}=0.0168 m^2/s	V_3=0.335 m/s Z_5=0.05 m q_3=0.0168 m^2/s

The hydraulic jump forms at X=2.25 m. While a series of small rollers develop on the surface, the downstream water surface remains smooth (Figure 82) with no oscillations after the jump. These features categorize the jump as a weak jump type (Liakopoulos, 2019). The Froude number of the jump is 2.25, confirming its classification as a weak jump based on its Froude number (Fr_1 = 2.25). Comparison between simulation results and the standard 1D analysis of the hydraulic jump are presented in Table 21.

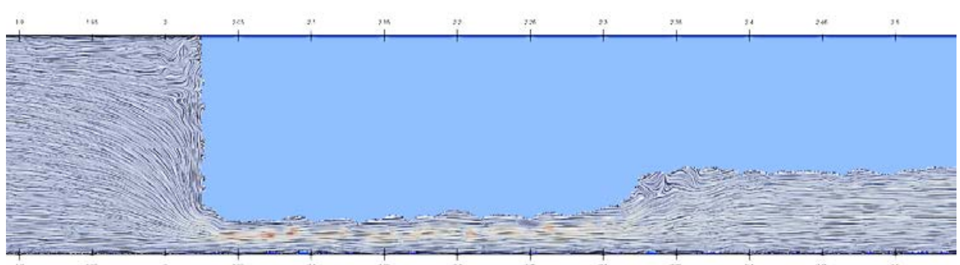


Figure 82. Side view. Y=0.75 m half channel's width. Characteristics of the hydraulic jump. T=200 sec.

Table 21. Comparison of SPH results with experimental-empirical estimations.

Case 3	SPH (3D)	Theoretical (1D)	Rel. diff %
y_2	0.05	0.0493	1.4 %

The relative difference in this case is 1.4% which underscores the better effect of the Laminar+SPS viscosity treatment at the simulation.

Yoosefdoost et al. (2022) conducted experimental studies to examine the steady-in-the-mean flow through a sluice gate, focusing on how the upstream and downstream water depths influence the flow behavior. Their primary goal was to determine the conditions for free, submerged, and partially submerged flow downstream of the sluice gate, as well as to calculate the discharge coefficient under various experimental setups. Notably, the channel geometry used in their experiments matches the configuration employed in the numerical simulations presented in this study. They explored a range of flow rates, gate openings, and weir heights, covering a wide spectrum of sluice gate operating scenarios, including both free and submerged hydraulic jumps.

However, in this study, we specifically focused on scenarios involving free hydraulic jumps. The simulated conditions included a flow rate of Q=2.5 L/s, a gate opening of Y_G=25 mm, and a weir height of H_{weir}=25 mm. The results from the SPH simulations showed good consistency with both the experimental data reported by Yoosefdoost et al., and established empirical relationships (Swamee, 1992), as illustrated in Figure 83.

The simulation labeled as "DBC" utilized DBCs with a VB value of 1. The chosen turbulence model was the Laminar + SPS, with all other simulation parameters kept consistent with the previous two cases.

It was observed that the DBC simulation exhibited less accurate agreement with the experimental data compared to other methods. This outcome was expected since standard DBCs have known limitations related to density stabilization near solid boundaries, often leading to pressure oscillations close to the walls. The mDBC approach used in this study proved to be more effective, highlighting its superiority in maintaining boundary stability.

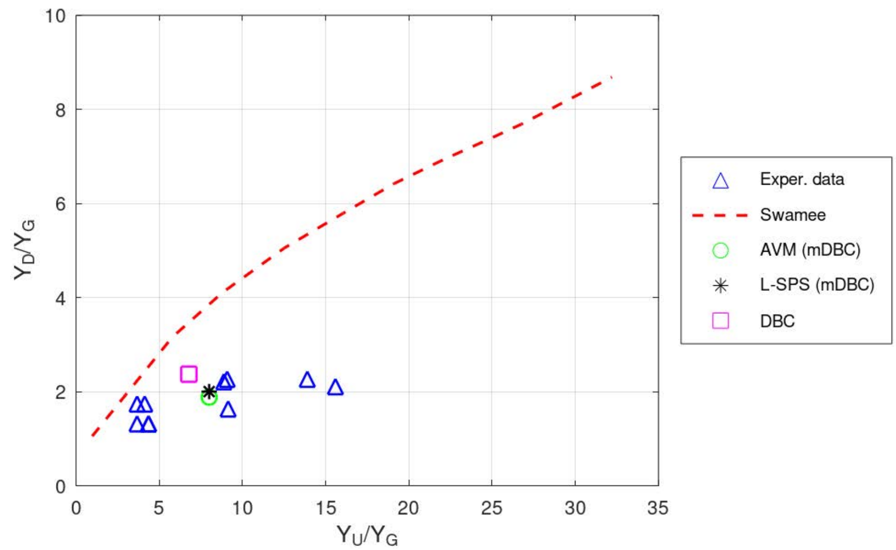


Figure 83. Comparison of SPH results with experimental results (Yoosefdoost et al., 2022) and Swamee empirical relations.

Previous simulation results with DBC solid BCs in the same geometry can be found in Appendix C (Figures C6 to C16) and more analytical discussion of the results in the work of (Chatzoglou and Liakopoulos, 2023)

4 Modifying the DualSPHysics Code-Developing the SPH method

This chapter presents the development and implementation of the Dynamic Smagorinsky turbulence model within the DualSPHysics computational framework. The motivation for choosing this adaptive turbulence modeling approach is explained, followed by detailed descriptions of code modifications, implementation steps, and subsequent validation through carefully selected test cases, highlighting the model's improvements over existing approaches.

4.1 DualSPHysics source code

The implementation of the Dynamic Smagorinsky model in DualSPHysics required modifications to specific source code files. The implementation workflow involved four key steps:

1. Construct a Wendland kernel with a smoothing length two times h.
2. Modifying the Standard Smagorinsky framework to include test-filtered strain rate tensors.
3. Adding memory allocation and initialization for new variables and arrays.
4. Updating functions to calculate the Dynamic Smagorinsky coefficient dynamically, based on local flow conditions.

To make modifications to the source code, a specific process should be followed. In most cases, only a few scripts require changes. This unit outlines the workflow through the process effectively. In Figure 84, the simulation flow chart is depicted.

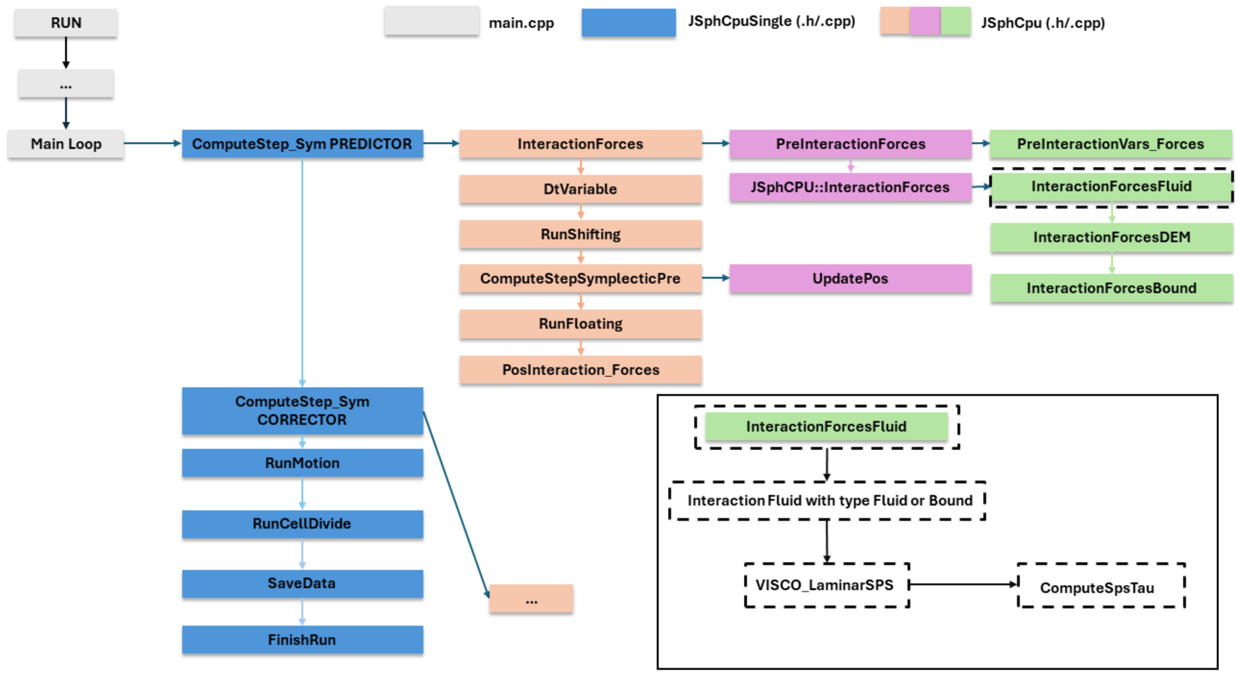


Figure 84. Overview of the Symplectic integration scheme used for time-stepping in the DualSPHysics code, illustrating the computational steps involved in executing simulations on CPUs.

More specifically, for the modifications made in this study the basic source files that need to be modified are JSph.h/cpp, JSphCpu.h/cpp and JSphCpuSingle.h/cpp. More information about each file can be found in Table 22 and the workflow diagram is depicted in Figure 85.

Table 22. List of source files modified in the DualSPHysics code for the implementation of the Dynamic Smagorinsky model, with descriptions of their roles.

Source file	Description
main.cpp	The main file that executes the code on CPU (or GPU)
JSph.cpp	Implements the class that defines all the attributes and functions for both CPU and GPU simulations
JSph.h	Declares the class that defines attributes and functions
JSphCpu.cpp/.h	Implements the class that defines attributes and functions only used by CPU simulations.
JSphCpuSingle.cpp/.h	Implement attributes and functions used to arrange CPU simulations

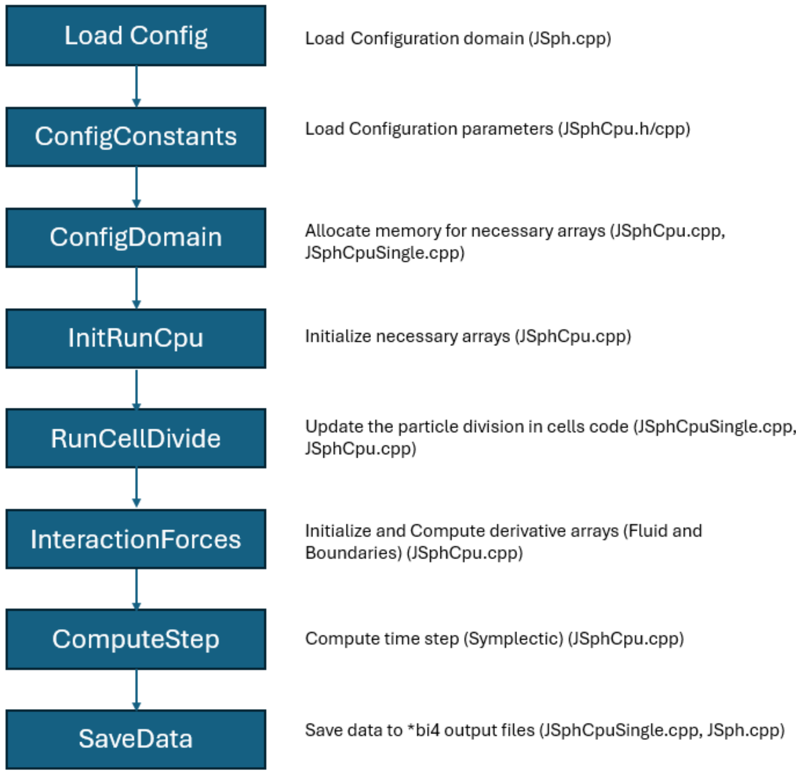


Figure 85. Computational workflow diagram depicting the steps involved in the implementation of the Dynamic Smagorinsky model within the DualSPHysics framework.

4.2 Implementation of the Dynamic Smagorinsky model into DualSPHysics code (LES-SPH)

The Dynamic Smagorinsky turbulence model was selected due to its ability to dynamically adjust turbulence coefficients based on local flow conditions, eliminating the need for empirical calibration typically required in the standard Smagorinsky model. This adaptability enhances accuracy in capturing turbulent flows, particularly near boundaries and in regions of complex fluid dynamics.

Favre-averaging is essential for addressing compressibility in weakly compressible SPH as described by Dalrymple and Rogers (2006). In their approach, the eddy viscosity assumption is employed to model the Sub-Particle Scale (SPS) stress tensor. Using Einstein notation, the stress components in the coordinate directions i and j are calculated as:

$$\frac{\tau_{ij}}{\rho} = \nu_\tau\left(2S_{ij} - \frac{2}{3}k\delta_{ij}\right) - \frac{2}{2}C_I\Delta^2\delta_{ij}|S_{ij}|^2 \tag{4-1}$$

where ν_τ is the turbulent eddy viscosity and k the turbulent kinetic energy per unit mass. Δ is the initial interparticle distance. C_I is Blin's constant (Blin et al., 2003) and takes the value $C_I = 0.0066$. S_{ij} is the Favre-filtered strain rate tensor

$$S_{ij} = -\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{4-2}$$

In the standard Smagorinsky model (Smagorinsky, 1963) the eddy viscosity is given by

$$\nu_\tau = (C_s\Delta)^2|\bar{S}| \tag{4-3}$$

where the local strain rate is given by

$$|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{4-4}$$

and the standard Smagorinsky constant takes the value C_s=0.12. The overbar denotes that a spatial filter has been applied, meaning that only the large-scale structures are solved directly, while the sub grid-scale (SGS) effects are modelled (Versteeg and Malalasekera, 1995).

The fundamental concept of the Dynamic Smagorinsky model is that the Smagorinsky constant is treated as a variable, parameterized by two distinct filters (Khani and Waite, 2015). The first filter is the grid filter, which, as previously mentioned, is inherently satisfied by the kernel function. The second is the test filter, which can be satisfied by the Wendland kernel function choosing different kernel length. Consequently, the smoothing length h is applied for the grid filter (denoted as subscript g), and the length 2h is utilized for the test filter (denoted by subscript t). A variable filtered by the grid and test filters is represented accordingly as $\overline{(\cdot)}$ and $\widetilde{(\cdot)}$ respectively.

Germano's identity (Germano et al., 1991) is used to calculate the sub-particle scale stress tensor differences L_{ab}

$$L_{ab} = 2C_d\Delta^2 M_{ab} \tag{4-5}$$

$$M_{ab} = \left(\frac{{h_t}^2}{{h_g}^2}\right)|\tilde{\bar{S}}|\tilde{\bar{S}}_{ab} - (\widetilde{\overline{|S|S_{ab}}}) \tag{4-6}$$

while the SPH discretization of eqs (4-7, 4-8) is

$$\tilde{\bar{S}}_{ab} = \sum_j \frac{m_j}{\rho_j} (\bar{S}_{ab,j}) W_{ij,t} \tag{4-7}$$

$$|\tilde{\bar{S}}| = \sqrt{2\tilde{\bar{S}}_{ab}\tilde{\bar{S}}_{ab}} \tag{4-8}$$

$\tilde{\bar{S}}_{ab}$ represents the strain rate tensor after the grid and test filtering. The approximation for the filtered product of the strain rate magnitude in Eq. 4-6 can be expressed as

$$\widetilde{\overline{|S|}\bar{S}_{ab}} = \sum_j \frac{m_j}{\rho_j} (\overline{|S|}\bar{S}_{ab,j}) W_{ij,t} \tag{4-9}$$

while the dynamic Smagorinsky coefficient C_d (Choi and Kim, 2023) is computed as

$$C_d \Delta^2 = \left[\frac{1}{2} \frac{L_{ab}\, M_{ab}}{M_{ab} M_{ab}}\right]_V \tag{4-10}$$

In Eq. 4-10 $[\cdot]_V$ indicates the volume averaging operator.

As mentioned before, the main idea behind the Dynamic Smagorinsky is that in the computation of the dynamic Smagorinsky coefficient we used filtered strain rate tensor not only with the grid filter but also with test filter (see Eqs 2-60 to 2-63 and 4-5 to 4-10). Accordingly, we follow the path of the already existing Standard Smagorinsky (Laminar+SPS) adding, where necessary, the new elements, equations, and variables. Part of the computation of the Sub-Particle-Scale (SPS) stress tensor is presented in Algorithm 2.

***Algorithm 1**. Part of the DualSPHysics code for the computation of the SPS stress tensor. Source file: JSphCpu.cpp.*

```
1.   /// Computes sub-particle stress tensor (Tau) for SPS turbulence model.
void JSphCpu::ComputeSpsTau(unsigned n,unsigned pini,const tfloat4 *velrhop,const tsymatrix3f *gradvel,tsymatrix3f *tau)const{

  const int pfin=int(pini+n);

  #ifdef OMP_USE

    #pragma omp parallel for schedule (static)

  #endif
```

```
  for(int p=int(pini);p<pfin;p++){
    const tsymatrix3f gradvel=SpsGradvelc[p];
    const float pow1=gradvel.xx*gradvel.xx + gradvel.yy*gradvel.yy + gradvel.zz*gradvel.zz;
    const float prr=pow1+pow1 + gradvel.xy*gradvel.xy + gradvel.xz*gradvel.xz + gradvel.yz*gradvel.yz;
    const float visc_sps=SpsSmag*sqrt(prr);
    const float div_u=gradvel.xx+gradvel.yy+gradvel.zz;
    const float sps_k=(2.0f/3.0f)*visc_sps*div_u;
    const float sps_blin=SpsBlin*prr;
    const float sumsps=-(sps_k+sps_blin);
    const float twovisc_sps=(visc_sps+visc_sps);
    const float one_rho2=1.0f/velrhop[p].w;
    tau[p].xx=one_rho2*(twovisc_sps*gradvel.xx +sumsps);
    tau[p].xy=one_rho2*(visc_sps   *gradvel.xy);
    tau[p].xz=one_rho2*(visc_sps   *gradvel.xz);
    tau[p].yy=one_rho2*(twovisc_sps*gradvel.yy +sumsps);
    tau[p].yz=one_rho2*(visc_sps   *gradvel.yz);
    tau[p].zz=one_rho2*(twovisc_sps*gradvel.zz +sumsps);
  }
}
```

As shown in Algorithm 2, the strain rate tensor is computed using the "*gradvel*" (grad velocity) where x, y, z indicates the respective directions. This computation follows Einstein summation notation. Due to the tensor's symmetry, the symmetric terms are not shown.

The first step in the implementation is to declare a new parameter for the gradvel, we named it d_gradvel in which the kernel of the computation should have a smoothing length of 2h instead of h (see section 2). After the declaration, memory should be allocated for specific arrays, and these arrays should be initialized (see Algorithm 3). Following the Standard Smagorinsky formulation, which already existed in the source code, we declare the dgradvelp1 (where p1 is a fluid particle), see Algorithms 4, 5. The terms L_{ab}, M_{ab} are derived similarly to Eqs (4-5 to 4-10) which will be used for the computation of the new visco_sps (Dynamic Smagorinsky coefficient) according to Eq. 4-10.

Algorithm 2*. Initialization of the dgradvelp1 matrix, according to the gradvelp1 matrix already existed.*

```
for(int p1=int(pinit);p1<pfin;p1++){
  float visc=0,arp1=0,deltap1=0;
  tfloat3 acep1=TFloat3(0);
  tsymatrix3f gradvelp1={0,0,0,0,0,0};
  tsymatrix3f dgradvelp1{0,0,0,0,0,0};
```

Algorithm 3. *Implementation of the dgravel.*

```
const int th=omp_get_thread_num();
    if(visc>viscth[th*OMP_STRIDE])viscth[th*OMP_STRIDE]=visc;
    if(tvisco==VISCO_LaminarSPS){
     gradvel[p1].xx+=gradvelp1.xx;
     gradvel[p1].xy+=gradvelp1.xy;
     gradvel[p1].xz+=gradvelp1.xz;
     gradvel[p1].yy+=gradvelp1.yy;
     gradvel[p1].yz+=gradvelp1.yz;
     gradvel[p1].zz+=gradvelp1.zz;

     dgradvel[p1].xx+=dgradvelp1.xx;   //Implementation of the dgradvel which is used
for the computation of Sij
     dgradvel[p1].xy+=dgradvelp1.xy;
     dgradvel[p1].xz+=dgradvelp1.xz;
     dgradvel[p1].yy+=dgradvelp1.yy;
     dgradvel[p1].yz+=dgradvelp1.yz;
     dgradvel[p1].zz+=dgradvelp1.zz;

    }
```

***Algorithm 4**. Velocity gradients computation. Where p1, p2 are fluid particles, frx;fry;frz are kernel gradients. Accordingly, the dfrx is the kernel derivative with a smoothing length 2h instead of h. frx=fac*drx,fry=fac*dry,frz=fac*drz; //-Gradients.*

```
//-Velocity gradients.
        if(!ftp1){//-When p1 is a fluid particle.
         const float volp2=-massp2/velrhop2.w;
                    float  dv=dvx*volp2;  gradvelp1.xx+=dv*frx;  gradvelp1.xy+=dv*fry;
gradvelp1.xz+=dv*frz;
                          dv=dvy*volp2;  gradvelp1.xy+=dv*frx;  gradvelp1.yy+=dv*fry;
gradvelp1.yz+=dv*frz;
                          dv=dvz*volp2;  gradvelp1.xz+=dv*frx;  gradvelp1.yz+=dv*fry;
gradvelp1.zz+=dv*frz;

        tsymatrix3f dgradvelp1;
          float dv_d=dvx*volp2; dgradvelp1.xx+=dv_d*dfrx; dgradvelp1.xy+=dv_d*dfry;
dgradvelp1.xz+=dv_d*dfrz;
               dv_d=dvy*volp2; dgradvelp1.xy+=dv_d*dfrx; dgradvelp1.yy+=dv_d*dfry;
dgradvelp1.yz+=dv_d*dfrz;
               dv_d=dvz*volp2; dgradvelp1.xz+=dv_d*dfrx; dgradvelp1.yz+=dv_d*dfry;
dgradvelp1.zz+=dv_d*dfrz;

//-To compute tau terms, we assume that gradvel.xy=gradvel.dudy+gradvel.dvdx,
//gradvel.xz=gradvel.dudz+gradvel.dwdx, gradvel.yz=gradvel.dvdz+gradvel.dwdy
//-so only 6 elements are needed instead of 3x3.
```

Once all the necessary parameters and variables have been declared, we introduce the new visco_sps parameter, referred as “Cd”, following the steps outlined in Algorithm 1. This process involves several key steps: declaring the new variable, allocating the required arrays, and initializing them properly. Additionally, it is crucial to ensure that the new variables are included in the declarations of all relevant functions to maintain consistency and avoid compilation issues. More details on the implemented code can be found in Appendix D.

To assess the implementation’s effectiveness two test cases are considered: the 3D straight rectangular open channel flow and the 3D square-cross-section closed channel. It should be noted that LES simulations should be conducted in three dimensions (Violeau, 2012). The standard Smagorinsky model and the implemented Dynamic Smagorinsky model are evaluated exclusively in 3D channels, using PBCs in the main flow direction to reduce computational cost. All simulations are performed on ARIS. supercomputer clusters, leveraging the code's parallelization capabilities.

4.3 Test Cases

4.3.1 3D Straight Open Channel

The first test case examines the problem as studied by Violeau and Issa (2007), who applied the standard Smagorinsky model. This enables a direct comparison between the SPH results of this study those obtained by Violeau and Issa (2007). The simulation model set up is depicted in Figure 86 and simulation's parameters are presented in Table 23.

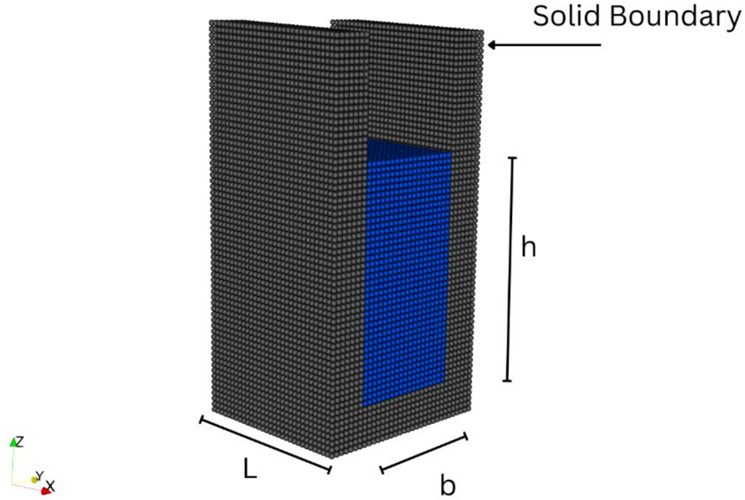


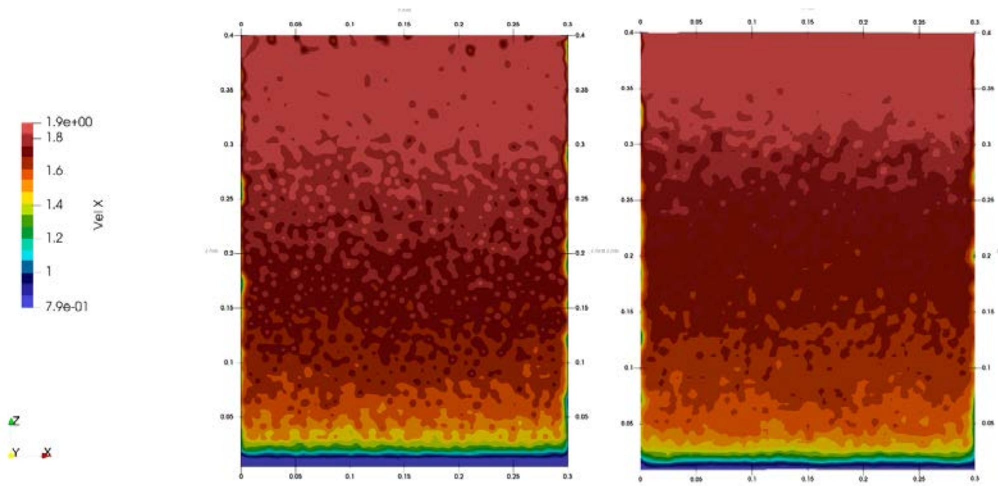
***Figure 86*.** Simulation model set up for the straight 3D open channel of orthogonal cross section. Periodic BCs are imposed in the streamwise direction.

Table 23. Straight 3D open channel. Simulation Parameters. Np is the number of particles, b is the channel's width, h is the water depth and dp is the initial interparticle distance.

Np	47926
L	0.3 m
b	0.2 m
h	0.4 m
dp	0.01
Streamwise-BCs	Periodic
Liquid/Solid-BCs	mDBCs

As shown in Figure 87, the color plot of the X-velocity component exhibits similar characteristics in both snapshots, with both velocity fields displaying typical turbulent behavior. However, Figure 87b, which is based on the in-house implemented modified Dynamic Smagorinsky model, demonstrates a more uniform and smoother distribution of velocity, indicating improved resolution of turbulent structures compared to Figure

87a, which is based on the standard Smagorinsky model. Additionally, Figure 87b shows a more gradual velocity gradient near the bed, suggesting better representation of the boundary layer. From a qualitative perspective, Figure 87b provides better quality results in capturing the turbulent flow structures.



(a) (b)

Figure 87. 3D Open channel. X-component of the velocity. Sideview snapshot at T=200 sec, y=0.1 m (half channel's width). a) original (Standard Smag.), b) modified (Dynamic Smag.).

In both SPH simulations the mass conservations law is satisfied, and no fluid particles penetrate the solid walls. Mean velocity is $\bar{V} = 1.345\, m/s$, leading to a bulk Reynolds number Re= $\frac{\bar{V}h}{\nu}$ =538000, friction Reynolds number Re_τ = $\frac{u_* h}{\nu}$ =20000 and a Froude number Fr= 0.68.

To better understand the influence of the modification, we compare the axial velocity profiles for both simulations with the logarithmic law

$$u^+ = \frac{1}{\kappa} ln z^+ + B \qquad \textit{(4-11)}$$

where u^{+}, z^{+} are the x-dimensionless velocity and distance from the channel bottom, respectively. Here, $u^{+}=u/u_{*}$ and $z^{+}=(zu_{*})/\nu$, u_{*} is the friction velocity, κ is the Karman constant and B an additive constant.Results are in relatively good agreement with the log law in most of the channels' depth (Figure 88a). However, as can be seen in Figure 88b, the implemented Dynamic Smagorinsky model has better agreement with the log-law closer to the solid wall. The Standard Smagorinsky (Laminar+SPS) and results from (Violeau and Issa, 2007) over predict the velocity close to the solid boundary.

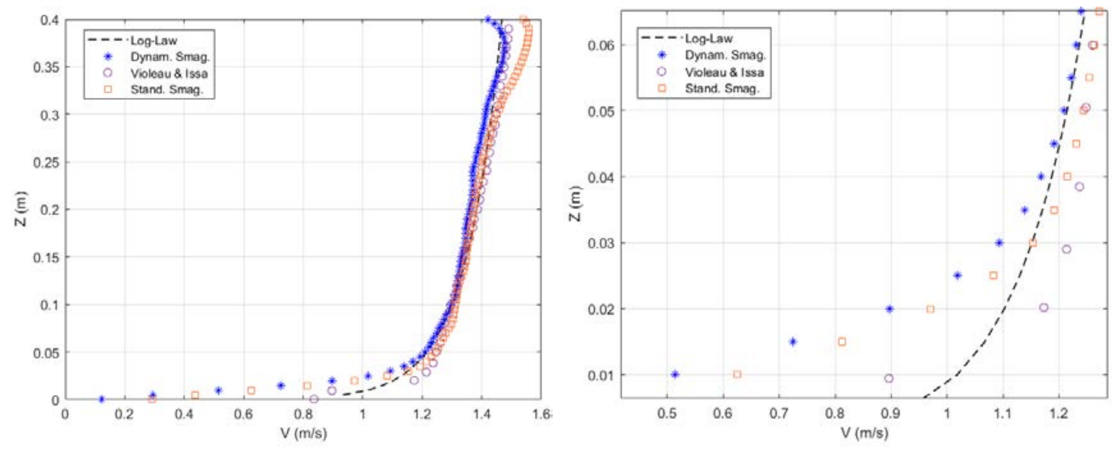


Figure 88. Spatially Averaged dimensional velocity profile, y=0.1 m (half width), u*=0.05 m/s. a) Velocity profile b) The near-wall region (z = 0.01 m to z = 0.065 m), where the Dynamic Smagorinsky model demonstrates improved adherence to the log-law (κ=0.4, B=5.2).

The Dynamic Smagorinsky model is expected to improve velocity prediction very close to the solid wall which is achieved up to a certain degree. Yet, the nature of the SPH and the physical gap between solid and fluid particles lead to inaccuracies at the fluid/solid interface. Even with the use of mDBCs the gap between fluid and solid is dp/2 (equals to 0.005 m in our cases). A possible solution would be a multiresolution approach for the fluid particles, which is not available in this version of the DualSPHysics code (v5.2), or the implementation of a more sophisticated wall function.

To make better comparison between the SPH velocity results and the log-law the dimensionless velocity and distance are plotted in semi-log axes. As can be seen in Figure 89, neither turbulence model could produce precise results very close to the solid boundary. Yet, the Dynamic Smagorinsky model shows much better agreement with the log-law than the Standard Smagorinsky model. It is important to note that this agreement is achieved with a relative low resolution (dp=0.01 m), indicating that the Dynamic Smagorinsky model can produce satisfactory results without the vast computational cost needed for the LES-SPH approach with the Standard Smagorinsky. A higher resolution (dp=0.005) was also tested for both simulations, and it did not lead to an enhancement in the results.

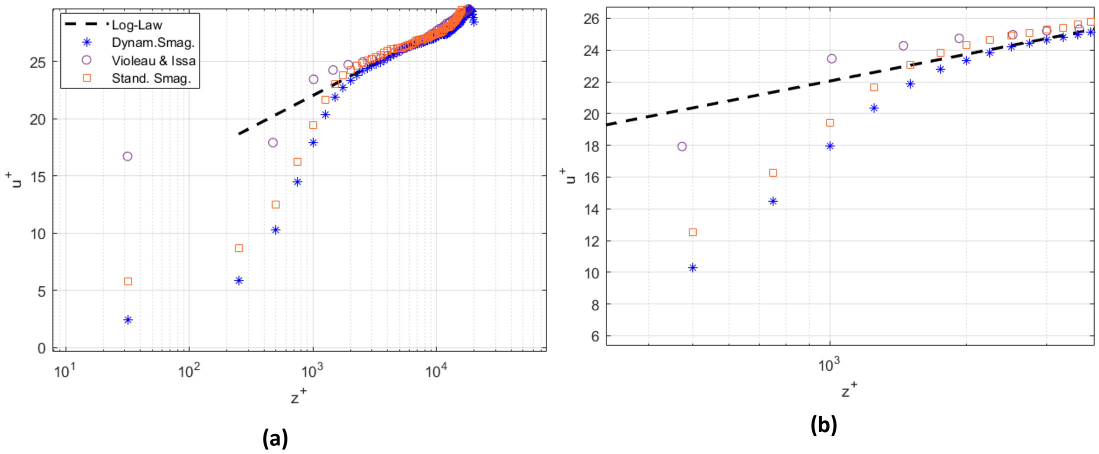


Figure 89. Semi-log plots of dimensionless velocity profiles. a) Semi-log plot of the full velocity profile. b) Near-wall region (0.01 ≤ z ≤ 0.065 m), highlighting the improved agreement of the Dynamic Smagorinsky model with the logarithmic law (κ=0.4, B=5.2).

4.3.2 3D Closed Channel

To further evaluate the implementation, a 3D confined flow test case is presented. Notably, there is limited literature on confined turbulent flows using the SPH method and even less on LES-SPH simulations for confined flows. Mayrhofer et al. (2015) employed a quasi SPH-DNS approach which demonstrated good agreement with reference data. However, in their LES-SPH approach (Re_τ=1000), the SPH simulation overpredicted streamwise velocity by approximately 20%. Therefore, we assessed our implementation in a similar case to a smaller computational domain to save computational time. The setup of the simulation model can is depicted in Figure 90, while its parameters are presented in Table 24.

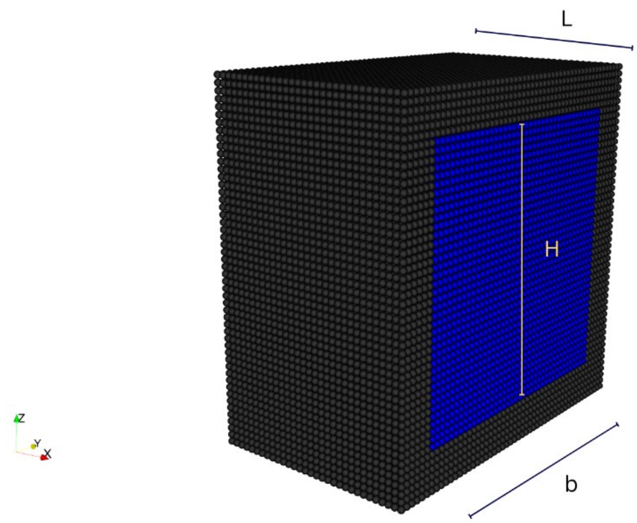


Figure 90. 3D closed channel. Simulation model configuration. Periodic BCs in the streamwise direction (x-direction).

Table 24. 3D closed channel. Simulation Parameters.

Np	80631
L	0.3 m
b	0.4 m
H	0.4 m
dp	0.01
Streamwise-BCs	Periodic
Liquid/Solid-BCs	mDBCs

The velocity field in this case is highly turbulent as expected. Details are shown in Figure 91, where color-coded snapshots of the X-component of velocity at the midplane (y=0.2 m) are presented at the steady-in-the-mean state. Figure 91a shows results from the Standard Smagorinsky and Figure 8b for the in-house implemented Dynamic Smagorinsky model. Both simulations capture the boundary layer, but the Dynamic Smagorinsky model (Figure 91b) provides more pronounced velocity fluctuations suggesting a more detailed representation of turbulence dynamics.

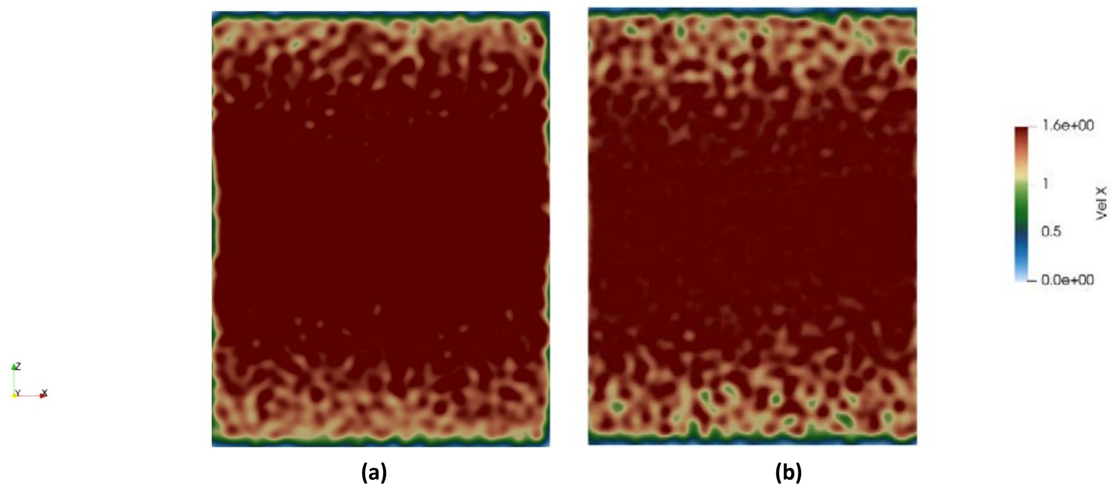


Figure 91. 3D closed channel. X-component of velocity. Sideview snapshot at T=200 sec, y=0.2 m (half channel's width). a) Original SPS (Standard Smagorinsky), b) Modified SPS (Dynamic Smagorinsky).

Both simulations satisfy the continuity equation and at a steady-in-the-mean state the mean velocity is $\bar{V}$=1.52 m/s resulting in a bulk Reynolds number Re= $\frac{\bar{V}D_h}{\nu}$ = 608000, $\bar{V}$ is the mean velocity of the flow, D_h the hydraulic diameter and ν the kinematic viscosity of the water.

In Figure 92, the dimensionless velocity vs distance-normal-to-the-wall profile as predicted by the Standard and Dynamic Smagorinsky models are compared with the log-law. The impact of the modified code is more evident in this group of simulations. The behavior very close to the solid wall is still inadequate for both models. As in the previous example, higher resolution cases were tested and did not produce the expected improvement.

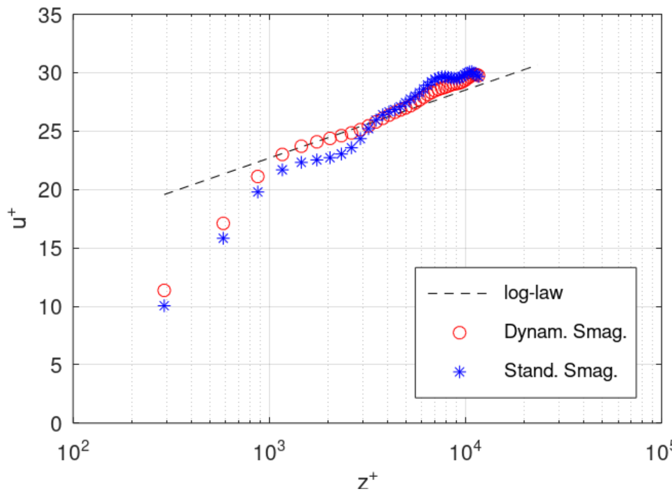


Figure 92. 3D closed channel. Non-dimensional velocity profile in inner law variables, (y=0.2 m).

Overall, the implemented Dynamic Smagorinsky model demonstrated significantly improved accuracy in predicting turbulent flow characteristics, especially in boundary layers and near-wall regions. This improvement substantially extends the DualSPHysics solver's capability to reliably simulate realistic, turbulent hydraulic flows without requiring extensive model calibration.

4.4 3D Open channel flow over vegetated bed.

This section demonstrates the practical applicability and robustness of the SPH method enhanced by the Dynamic Smagorinsky turbulence model and improved boundary conditions through detailed simulations of three-dimensional open-channel flow over submerged vegetation. This scenario, relevant to ecohydraulics and environmental engineering, presents complex fluid–structure interactions ideal for evaluating the improved SPH method.

Aquatic plants are commonly found in natural and manufactured water bodies such as rivers, lakes, and wetlands, where they significantly influence turbulent open-channel flows (Curran & Hession, 2013). Older studies noted their role in increasing flow resistance, raising water levels, and reducing channel capacity during floods, often prompting their removal to improve flow conveyance (Li et al., 2023; Nepf & Ghisalberti, 2008). However, aquatic vegetation also offers vital ecological benefits: enhancing water quality (Rao et al., 2018), supporting biodiversity (O'Hare, 2015), stabilizing banks, reducing erosion, and promoting sediment deposition (Qiu et al., 2024; Lorenz et al., 2015). By slowing flow through interception and absorption, plants reduce velocity and energy, aiding in soil retention and habitat growth. Understanding the interaction between vegetation and flow is essential for flood management and eco-hydraulic design (Zhou et al., 2016). Numerical modeling is a powerful tool for studying these dynamics, though accurately representing flow-vegetation interactions remains challenging. Studies by Kasiteropoulou et al. (2017), Qu and Yu (2020), and D'Ippolito et al. (2020) highlight the need for improved modeling of vegetation-induced resistance. Chatelain and Proust (2021) further explore how vegetation affects flow structure and turbulence. While traditional grid-based methods face limitations with complex geometries and free surfaces (Sibil, 2024; Jin & Zhang, 2022; Zeng et al., 2022), mesh-free approaches like SPH offer greater flexibility and accuracy (Violeau & Rogers, 2016; Roubtsova & Kahawita, 2006; Monaghan 1994). In this work, we adopt the SPH method to simulate turbulent open-channel flow with fully submerged vegetation, aiming to better capture the complex hydrodynamic interactions involved.

Developing a reliable vegetation model requires thoughtful selection of parameters such as plant geometry, height, spacing, and mechanical traits like flexibility or stiffness. To investigate the influence of vegetation on open-channel flow, Dunn et al. (1996) conducted controlled laboratory experiments to measure flow resistance caused by bed vegetation. These experiments were carried out in a flume with dimensions of 19.5 meters in length, 0.91 meters in width, and 0.61 meters in depth, and slopes ranging from 0.0036 to 0.0161. Both flexible and rigid cylinders were used to mimic vegetation, allowing analysis of their respective impacts on flow resistance. To validate our SPH model, we compared our simulation outcomes with both the experimental results of Dunn et al. (1996) and the numerical work of Kasiteropoulou et al. (2017). Our simulations adopted the same geometrical setup and represented vegetation using rigid cylinders. It should be noted that for computation purposes we select a small part of a very long channel as the computational domain. While this method omits the effect of plant flexibility, it establishes a foundational reference point for assessing vegetation-induced resistance in open-channel flow. In contrast to the experimental set-up, we applied PBCs along the streamwise direction in our simulations. This modification allowed us to reduce the length of the computational domain compared to the original 19.5-meter flume, thereby minimizing computational demands while maintaining the essential flow features. The driving force is the x-component of gravity acceleration. Table 25 summarizes the simulation geometry and the selected parameters used in our model. Figure 93 shows the computational domain, featuring a staggered arrangement of vegetation elements. This configuration mimics the irregularity of natural vegetation patterns, encouraging flow variability and generating turbulence structures typically seen in vegetated channels.

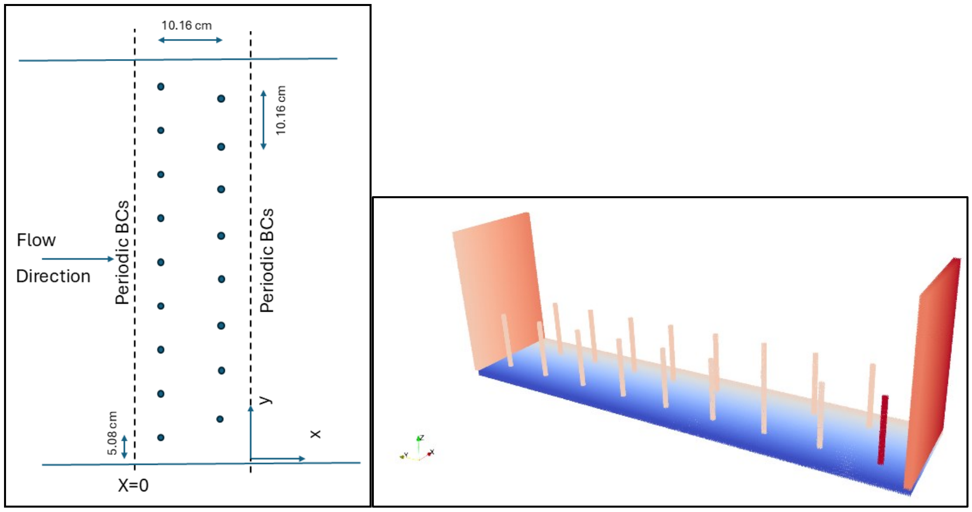


Figure 93. View of the computational domain and solid boundaries, showing the staggered cylinder arrangement. It should be noted that this is a small part of an infinite length channel.

Table 25. Simulation Parameters. L is the channel's length and b is the channel's width.

Geometry	**L (m)**	**B (m)**	**Flow depth (m)**	**cylinder radius (m)**	**cylinder height (m)**
	0.225	0.91	0.233	0.003175	0.118
Parameters	**Streamwise dir. BCs**	**Liquid/Solid BCs**	**Turbulence model**	**Viscosity (m^2/s)**	**Density (Kg/m^3)**
	Periodic	mDBCs	Laminar+SPS	0.000001	1000

One valuable tool for the simulation of these cylinders is the capability of the DualSPHysics code to be coupled with the project Chrono. Chrono is an open-source, high-performance library and suite of applications designed to support a wide range of use cases, including wheeled and tracked vehicles on deformable terrain, robotic systems, mechatronic applications, compliant mechanisms, and fluid-solid interaction phenomena. It allows for the creation of systems composed of both rigid and flexible/compliant components. Additionally, components can have complex three-dimensional geometries for accurate collision detection (El Rahi et al., 2024; Martinez-Estevez et al., 2023; Canelas et al., 2016).

After the flow reached steady-state conditions, the velocity fields are analyzed at various vertical and horizontal planes to investigate the impact of the submerged vegetation on the turbulent flow characteristics. This section presents a detailed discussion of the simulation results, highlighting the influence of the vegetated bed on flow velocity distributions, turbulence structures, and overall flow resistance. It should be noted that the use of Periodic BCs in the streamwise direction produces a channel of infinite length. For computational purposes we select a small part of that channel as computational domain.

In Figures 94-96 the x-component of velocity is depicted in a color-coded plane (x=constant). Figure 94 shows the x-velocity distribution at the inlet of the computational domain (x=0.01 m). In Figure 95 the plane presented refers to x=0.1 m. It is the region after the first row of cylinders and before the second (for the selected computational domain). Velocities are highly reduced in the location where the cylinders are located and even more between the cylinders of the same row. Regarding the free surface, we observe no big differences, compare to the planes beneath. Finally in Figure 96 is depicted the outstream computational domain (x=0.2 m). Compared to Figure 94, the velocity field is much more spatially heterogeneous, with clear patches of low velocity in the vegetated region.

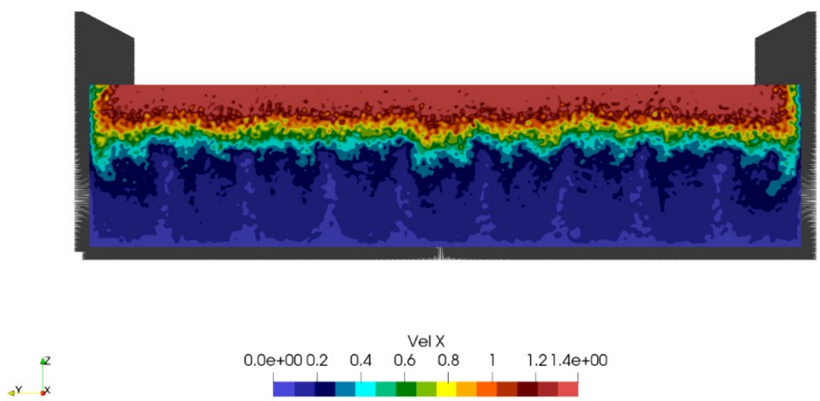


Figure 94. x-component of velocity field at x = 0.01 m. Streamwise distances are measured from the intel of the computational domain.

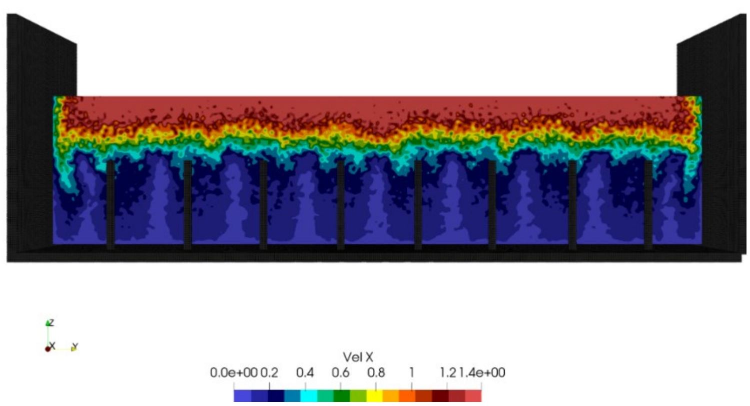


Figure 95. x-component of velocity field at x = 0.1 m. Velocity reductions and wake formation between the staggered cylinders are clearly visible.

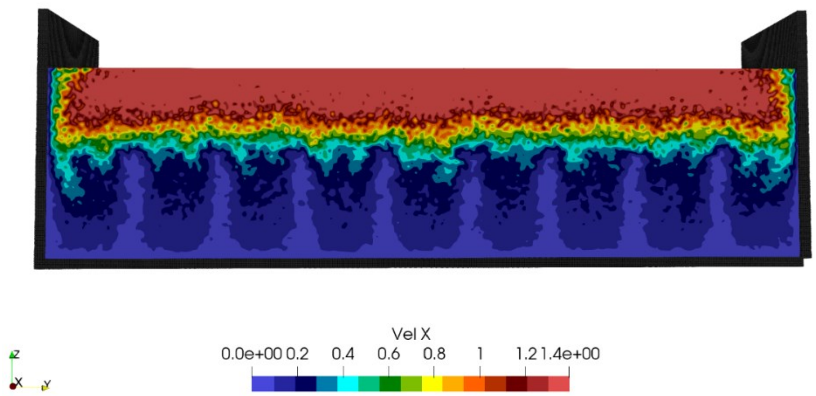


Figure 96. x-component of velocity field at x = 0.2 m, close to the computational domain outlet.

Horizontal planes are presented in Figures 97-102. In relatively low depth z=0.01 m (Figure 97) to the top of the cylinders z=0.118 m (Figure 100) and the free surface z=0.2 m (Figure 102). Near bed velocities (Figure 97) are low. This is critical for sediment transport and bed shear stress analysis. The reason is the reduced velocities which have as a result to not allow sediment particles to move. Thus, sediments are more likely to settle and remain stable, decreasing erosion and promoting deposition. As we move closer to the surface (Figure 98), we observe that wake structures begin to elongate and possibly merge, forming larger coherent low-velocity regions, which are even more visible in Figure 99. As the vertical elevation increases from the bed to the top pf the cylinders, wakes elongate, merge and clear vortices are formed (see Figure 100). The LES-SPH scheme used for this study gives a very clear representation of the roller-vortices which are formed due to the presence of the rigid cylinders (see Figure 101). The free surface depicted in Figure 102 has, as expected, higher velocities and is less affected by the presence of the vegetation than the planes referred to a lower water depth. The color scale is kept unchanged among the Figures in order to facilitate comparisons of water speed.

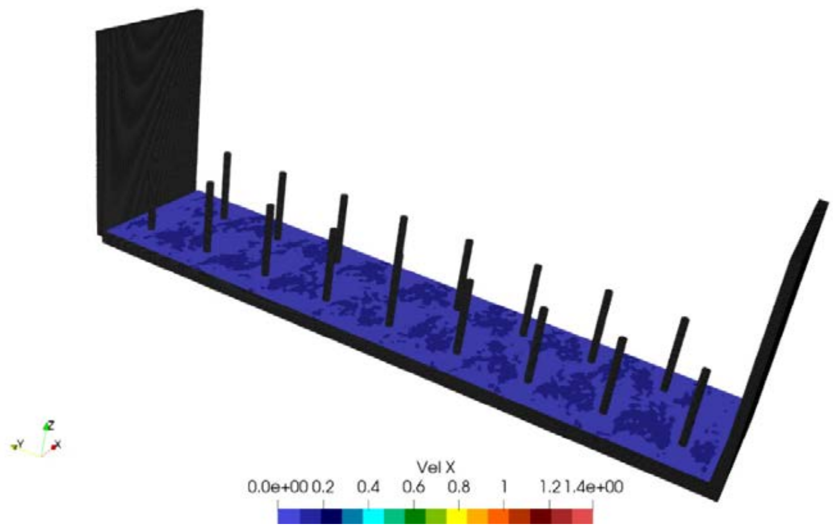


Figure 97. Plan view of x-component of velocity field at z = 0.01 m (near-bed).

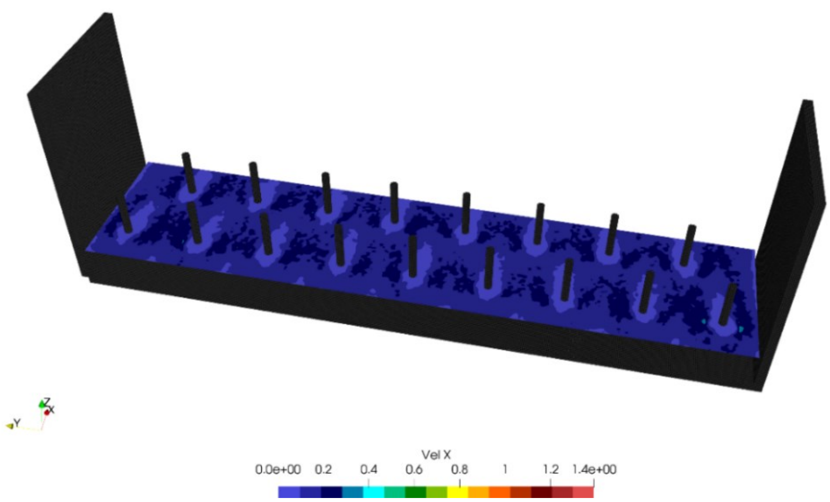


Figure 98. Plan view of x-component of velocity field at z = 0.05 m, showing elongated wake structures forming downstream of submerged vegetation elements.

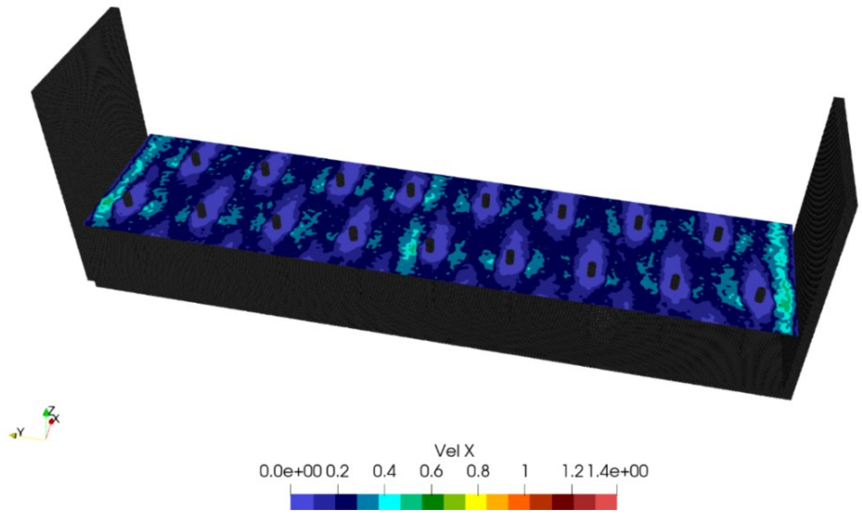


Figure 99. Plan view of x-component of velocity field at z = 0.1 m, highlighting increasing wake interactions and merging vortices as elevation increases.

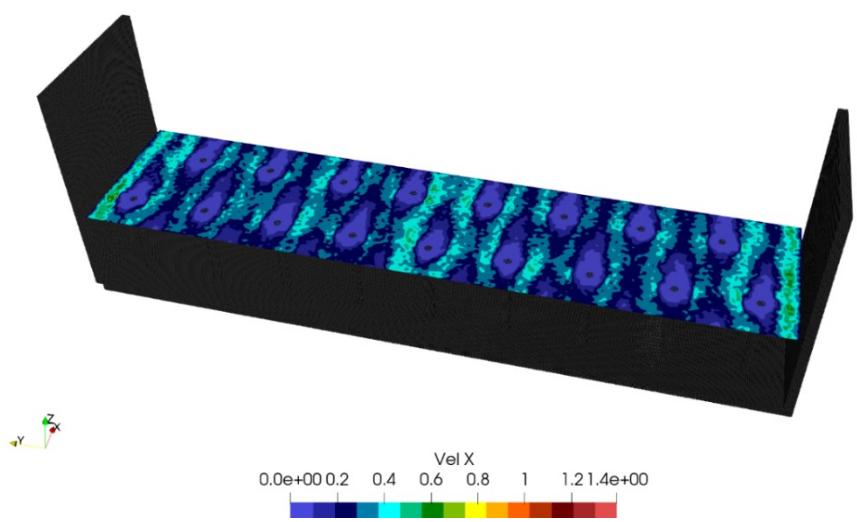


Figure 100. Plan view of X-component of velocity field at z = 0.118 m, at the top of the submerged cylinders.

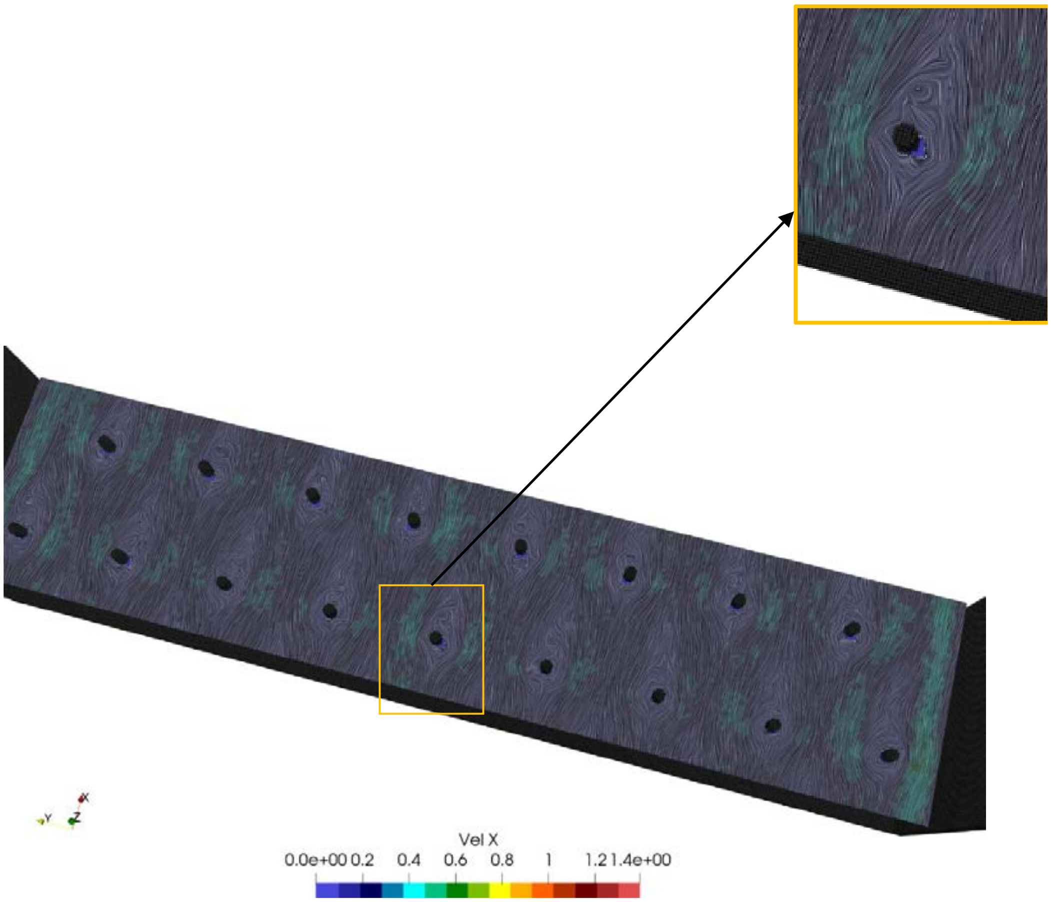


Figure 101. Streamlines and velocity magnitudes at z = 0.1 m, visualizing complex wake structures and turbulence induced by submerged vegetation. In the zoomed section of a random cylinder, we observe the vortex formation downstream of the cylinder.

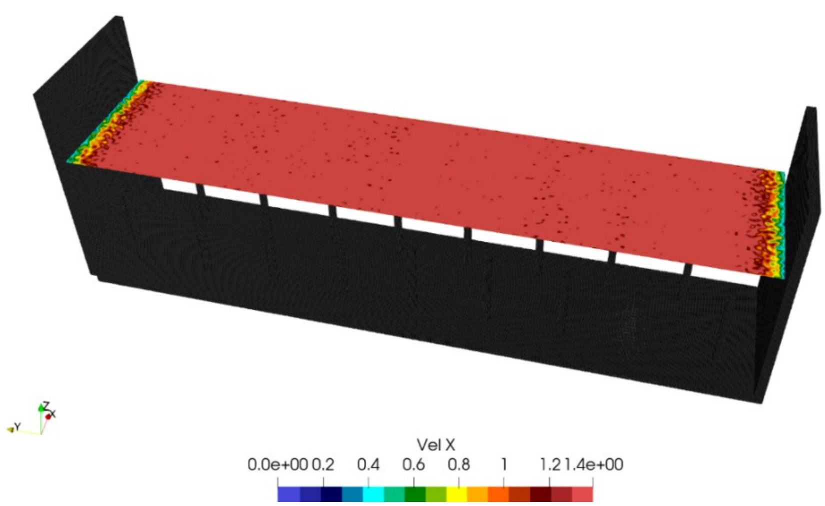
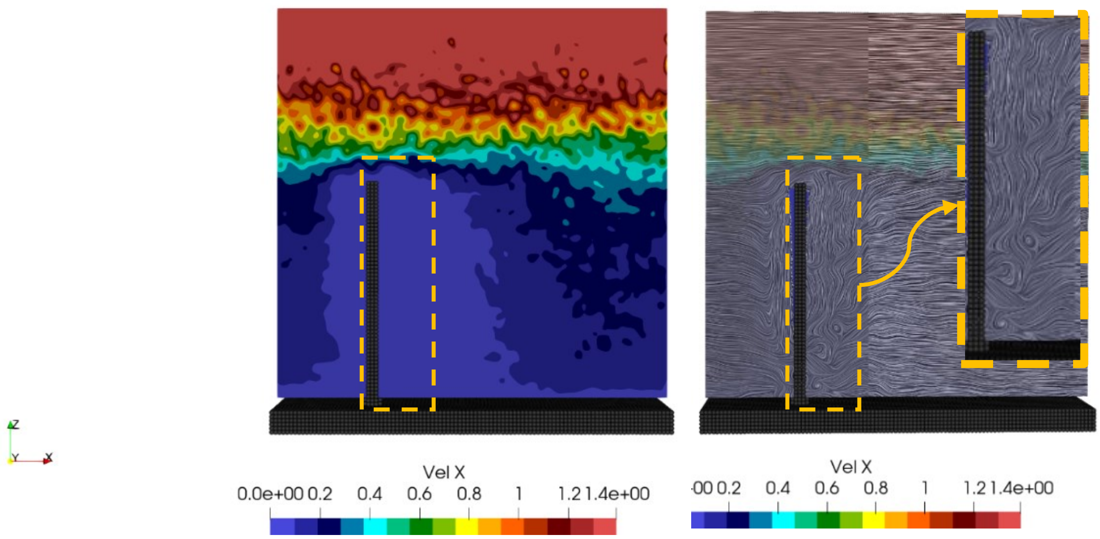


Figure 102. Plan view of X-component of velocity field at z = 0.233 m (free surface).

Finally, a vertical plane is depicted in Figure 103, at a location where a cylinder is placed (y=0.456 m). In Figure 103a we see the x-component of the velocity which gives information about the vertical velocity profile and the regions upstream and downstream of the cylinder. It is clear that the presence of the cylinder has a strong influence on the velocity. Streamlines presented in Figure 103b reveal vortices that are shaped after the rigid obstacle and possibly before.



(a) **(b)**

Figure 103. Vertical plane at y=0.456 m showing velocity field and streamlines around a vegetation cylinder. (a) x-component velocity distribution along the vertical plane, highlighting the flow, wake formation, and velocity gradients upstream and downstream of the cylinder. (b) Streamlines illustrate flow separation, recirculation, and vortex formation downstream the rigid cylinder and water distortion upstream.

In order though to compare the SPH results with experimental data and published data from other numerical tools we extracted the velocity profiles in several spanwise locations and when the flow reaches steady on the mean flow conditions, according to the work of (Kasiteropoulou et al., 2017).

As can be seen in Figure 104 the SPH velocity profile is in quite good agreement with the Kasiteropoulou et al. results and even better with the experimental data of Dunn et al. Moreover, the SPH model has lower relative errors when compared with the experimental data than (Kasiteropoulou et al., 2017) as is depicted in Figure 105 and Table 26. This makes clear that SPH is a very promising tool in the accurate simulation of complex open channel flows.

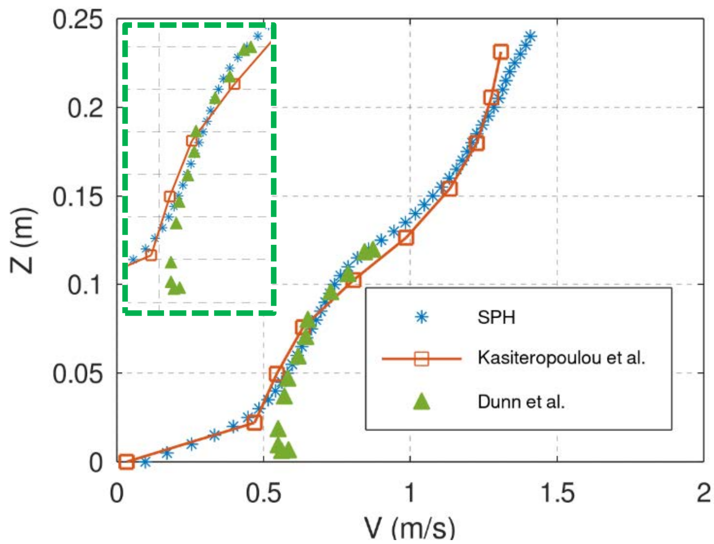


Figure 104. Comparison of velocity profiles extracted from SPH simulations, experiments (Dunn et al., 1996), and finite volume simulations (Kasiteropoulou et al., 2017). SPH results closely match experimental data, demonstrating the method's accuracy for vegetated open channel flows.

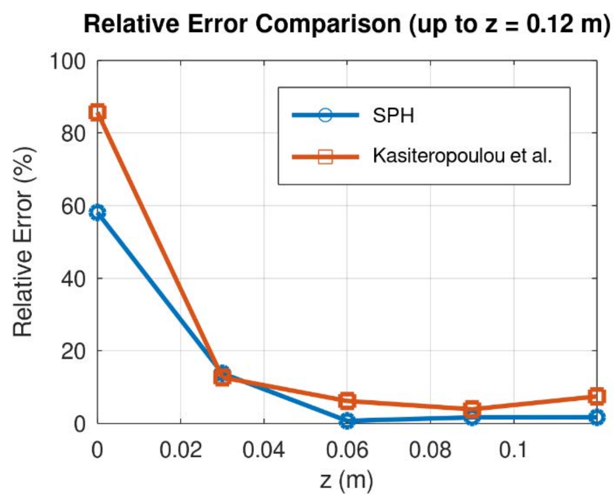


Figure 105. Comparison of relative errors (%) between the experimental data (Dunn et al., 1996), results from other conventional numerical methods (Kasiteropoulou et al., 2017) and the SPH results over the range 0 ≤ z ≤ 0.120 m. The plot illustrates the deviation of each dataset from the experimental data as a function of height (z), showing how the relative errors vary for each dataset.

Table 26. Relative velocity errors (%) at different heights, Z (m), comparing SPH, a conventional grid-based method and experimental data. SPH demonstrates improved accuracy with lower relative errors.

Z (m)	Error_conventional method (%)	Error_SPH (%)
0.03	12.6884	13.8112
0.06	6.1889	0.7082
0.09	3.9045	1.6866
0.12	7.4428	1.7003

In addition to the already good performance of SPH in such a complex case, we tested the Dynamic Smagorinsky implementation discussed in section 4. As in the simpler test cases of section 4, the additional computational time was slightly larger, yet acceptable. In Figure 106 the spatial averaged velocity profile of Kasiteropoulou et al., experimental results of Dunn et al., SPH simulation results using the Standard Smagorinsky model and SPH simulation results with our in-house implemented Dynamic Smagorinsky model are depicted. With a quick look there are not big differences between the SPH results retrieved from both models, yet with a closer look it is evident that the SPH-Dynamic Smagorinsky results have better predicted the behavior of the flow close to the free surface, very close to the solid walls and have better agreement to the experimental results. This is more evident in Figure 107 where a velocity profile from z=0-0.12 m is presented. Also, the relative velocity errors (Figure 108 and Table 27) make this assumption solid.

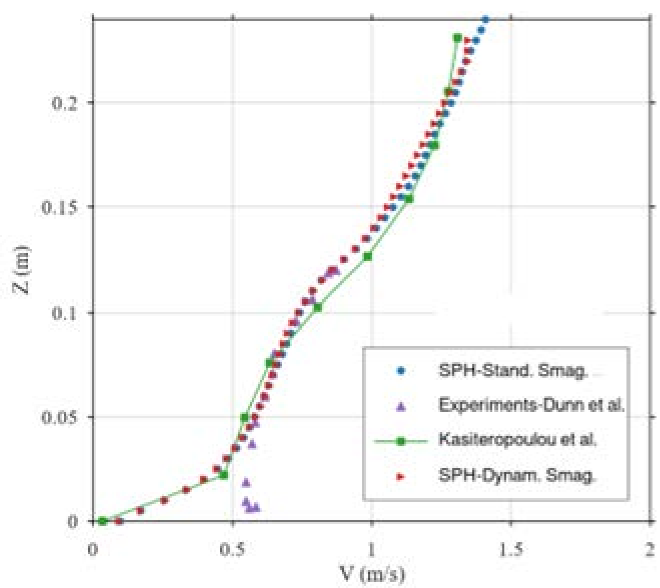


Figure 106. Comparison of velocity profiles extracted from SPH simulations with both Standard and Dynamic Smagorinsky models, experiments (Dunn et al., 1996), and finite

volume simulations (Kasiteropoulou et al., 2017). SPH-Dynamic Smagorinsky results slightly better match the experimental data, demonstrating the method's accuracy for vegetated open channel flows and the effectiveness of Dynamic Smagorinsky implementation described in section 4.

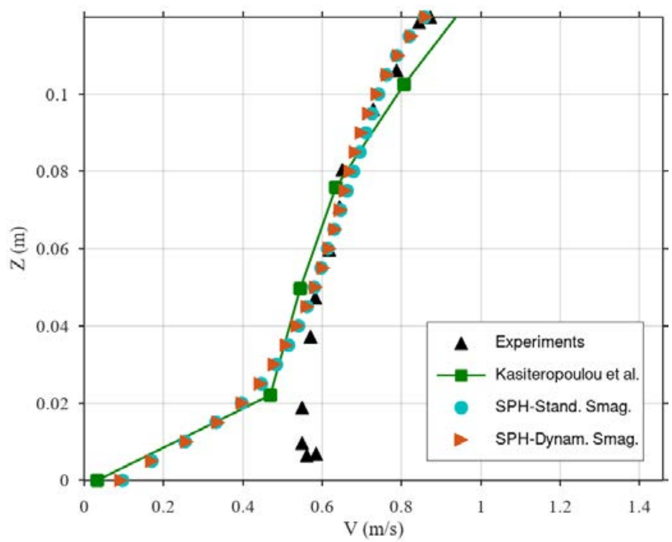


Figure 107. Velocity profile from z= 0 to 0.12 m (the top of the rigid cylinders).

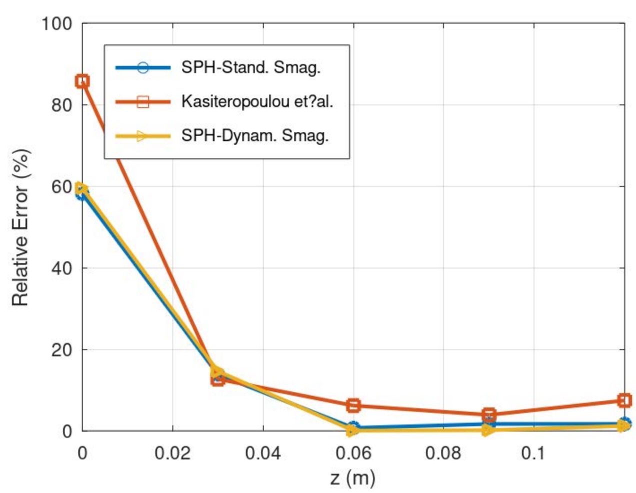


Figure 108. Comparison of relative errors (%) between the experimental data (Dunn et al., 1996), Kasiteropoulou et al. data, SPH- Standard Smagorinsky simulation results and SPH-Dynamic Smagorinsky simulation results over the range 0 ≤ z ≤ 0.120 m.

Table 27. Relative velocity errors (%) at different heights, Z (m), comparing SPH-Standard Smagorinsky, SPH-Dynamic Smagorinsky, a conventional grid-based method and experimental data.

Z (m)	Error_conventional method (%)	Error_SPH-Stand. Smag. (%)	Error_SPH-Dynam. Smag. (%)
0.03	12.6884	13.8112	14.61
0.06	6.1889	0.7082	0.07
0.09	3.9045	1.6866	0.16
0.12	7.4428	1.7003	1.21

Overall, the SPH method proved to be efficient in the simulations of such problems. In addition, the implementation of the Dynamic Smagorinsky model in the SPH framework proves again its effectiveness even in such a complex flow.

5 Conclusions

This chapter synthesizes the key results obtained throughout the thesis, highlighting the overall performance of SPH simulations, validated across various scenarios. Special emphasis is given to boundary condition performance, turbulence modeling, and the practical implications for hydraulic and environmental engineering. Furthermore, it provides an objective discussion of the limitations encountered, underscoring potential directions for future research.

5.1 General Overview

This thesis investigated the performance and development of the SPH method in simulating internal and open channel flows, encompassing a wide range of Reynolds numbers and flow regimes. Through the evaluation of various SPH formulations, comparison of two widely used open-source codes (LAMMPS and DualSPHysics), and the implementation of a Dynamic Smagorinsky turbulence model, this work aimed to assess the method's reliability and enhance its applicability in hydraulic and environmental engineering contexts. The findings confirm that SPH, while still under active development, demonstrates considerable potential as a flexible and meshless alternative to conventional Computational Fluid Dynamics (CFD) methods, particularly in scenarios involving free surfaces, complex geometries, or multiphase flows.

5.2 Validation of SPH for Internal Flows

The benchmark test cases of 2D and 3D laminar closed channel flows validated the ability of SPH to reproduce analytical solutions with high fidelity. As illustrated in Figures 8 and 18, the velocity profiles obtained from SPH simulations closely matched theoretical Poiseuille flow distributions for varying channel widths. The deviations were generally within 2–3%, as summarized in Table 4, indicating that SPH retains first-order spatial accuracy under appropriate parameterization.

Additionally, results confirmed the importance of kernel function selection and spatial resolution. Kernel performance near solid boundaries, especially in regions of contraction and expansion, had a noticeable impact on the accuracy and stability of the simulation. The Wendland kernel consistently offered improved particle ordering and minimized tensile instability compared to Gaussian and cubic spline alternatives. A kernel library was constructed in the context of this research where users can select the kernel of their preference and the library will show them the kernel, its first derivative plot and each kernel's strengths and limitations based on the authors of this research experience. More information can be found in Appendix E.

5.3 Boundary Condition Performance

Boundary condition treatment is a persistent challenge within SPH, as the method does not inherently enforce conditions at solid interfaces. This thesis tested and compared several implementations: Ghost Particles, Repulsive Particles, and Dynamic Particles within the LAMMPS code, and DBCs and mDBCs within DualSPHysics. There are other options available such as LUST BCs, yet the cases tested in this study have not been assessed.

Among these, the mDBC approach demonstrated the most robust performance. As depicted in Figure 5, the projection of ghost nodes into the fluid domain improved accuracy at wall boundaries and significantly reduced velocity and pressure anomalies. These improvements were most notable in geometrically complex cases such as expansions, contractions, and sluice gate flows (Figures 11–14). While the mDBC method introduces slightly increased computational cost due to ghost particle projection and interpolation, its benefits in numerical stability and physical accuracy justify its use in high-fidelity engineering simulations. One important parameter is the so called "ViscoBound factor." This parameter tunes the viscosity of the fluid at the solid boundary. Proper selection of this parameter can introduce the appropriate stress to the boundary layer zone (for turbulent flows).

5.4 Turbulent Flow Modeling Using the Dynamic Smagorinsky Model

A central contribution of this research is the implementation of a Dynamic Smagorinsky Large-Eddy Simulation (LES) model within the DualSPHysics framework. The model was developed to enhance SPH's capability in resolving turbulent flows without relying on empirically defined turbulence parameters.

Validation results, shown in Figures 88 and 89, revealed significant improvement in capturing the logarithmic region of the velocity profile near walls. In the range $0.01 \leq z \leq 0.065$ m, the dynamic model demonstrated superior alignment with the log-law compared to the standard Sub-Particle Scale (SPS) model. These improvements were further confirmed in 3D closed channel simulations (Figures 91 and 92), where the velocity distribution obtained with the dynamic model more closely matched the log-law.

This implementation validates the use of SPH-LES coupling, particularly for simulating wall-bounded turbulence, and suggests a promising pathway toward more adaptive and accurate turbulence modeling within meshless methods. However, further improvement very close to the solid walls is necessary. The methods particle nature and the inherited limitations from this need a more sophisticated approach given that even a high particle resolution will not produce the expected values.

5.5 Application-Oriented Assessment: Flow Over Vegetated Beds

To assess the practical capabilities of the SPH method, simulations of 3D open channel flows over submerged vegetation were conducted. These cases featured complex interactions between the flow and rigid cylindrical obstacles, presenting a realistic environmental hydraulics scenario. As shown in Figures 94 to 103, SPH successfully captured wake development, flow separation, and vortex structures. Velocity distribution at various vertical and longitudinal planes confirmed the method's ability to represent spatial variability within the flow. Comparative analysis with experimental and finite volume simulation results (Figure 104 and Table 26) showed relative errors generally below 8%, even in regions of intense turbulence and flow recirculation. These findings reinforce SPH's value as a viable CFD tool for riverine, ecological, and environmental flow applications. In addition, the in-house modified code descripted in section 4, have also been tested in this complex free surface case. The results obtained were in closer agreement with experimental data (see Figures 106, 107) than the conventional grid-based method and produce smaller relative error even from the SPH results with the Standard Smagorinsky model (see Figure 108 and Table 27). This proved once again the effectiveness of our implementation.

5.6 Limitations and Practical Considerations

Despite the promising results, the study identified several limitations that merit further attention:

- Computational cost: SPH simulations, especially in 3D and turbulent flow contexts, require significant computational resources. This is exacerbated by fine resolution requirements and the added complexity of turbulence models and coupling with other methods.
- Kernel support near free surfaces and boundaries: While mDBCs improve boundary treatment, kernel truncation still poses challenges in maintaining physical accuracy, particularly in multiphase or free-surface flows. In cases when the simulation parameters are not tuned effectively, tensile clumping or diffusion at the free surface may occur.
- Parameter tuning: Although the Dynamic Smagorinsky model reduces empirical dependence, other simulation parameters (e.g., smoothing length, artificial viscosity) still require calibration based on problem-specific characteristics. There is not a specific remedy which will be effective in every case. Also, from authors experience there are more than a few parameters that with proper handling can yield better results. An automated process to help the most effective parameters or a library with the most effective values depending on the case would be a huge asset for the method.

- Turbulence modeling: There is a current tendency to couple SPH method with LES and DNS in order to produce more accurate turbulent simulations. However, more sophisticated wall-functions, multiresolution approach and further development on the solid boundaries' imposition will improve the methods capabilities in capturing effectively turbulent structures in simulations with no need of vast resolution.

Addressing these limitations may involve integrating adaptive particle resolution, hybrid SPH-grid approaches, and GPU acceleration. Future work could also explore data-driven parameter optimization using machine learning to further automate and generalize SPH-based turbulence models (Sofos et al., 2024). Most of them are in the author's scope of future work which can be found analytically in subsection 7.2.

5.7 Overall Contributions

During the research, an in-depth study and application of the SPH method were carried out, with particular emphasis on the simulation of flows in both open and closed channels. This was achieved through two computational packages (codes): LAMMPS and DualSPHysics. The first code is used for simulating Molecular Dynamics (MD) problems and is compatible with SPH simulations. The second is specifically focused on SPH simulations, which is why it was selected for modification.

Specifically:

- Simulation models for flow in microchannels were developed.
- Models for simple flow examples (Poiseuille flow, laminar open flow, etc.) were developed.
- Flow simulations were conducted under a sluice gate and over a weir, investigating the formation of hydraulic jumps and the dynamics of the flow. Emphasis was placed on the correct implementation of boundary conditions.
- Turbulent flow simulation models were developed.
- Hydraulic jump simulations were performed using Open Boundary Conditions.
- Turbulent flow simulations were carried out in 3D open channels with vegetation on the bed.
- A comparison of the results was made with experimental data and literature results, confirming the reliability of the simulations.

This thesis also contributes to the advancement of SPH modeling in several key areas:

- It validated SPH for internal and open channel flows across multiple regimes using both LAMMPS and DualSPHysics codes.
- It highlighted the effectiveness of mDBC in improving boundary condition treatment, particularly for complex geometries.

- It implemented the Dynamic Smagorinsky turbulence model into DualSPHysics and demonstrated its superior performance over traditional SPS models.
- It applied SPH to realistic environmental scenarios, such as flow over vegetated beds, establishing its practical utility for engineering applications.

Taken together, these contributions provide a robust framework for further developing SPH as a reliable, accurate, and versatile CFD tool in hydraulic and environmental engineering.

In conclusion, this thesis focused on the comprehensive evaluation, validation, and enhancement of the SPH method as a computational tool for fluid flow simulation in hydraulic and environmental engineering applications. Motivated by the increasing demand for meshless, adaptive solvers capable of handling free-surface flows, complex geometries, and turbulent regimes, the study addressed both fundamental aspects of the SPH method and its practical application.

The first part of the thesis reviewed the theoretical framework of SPH, highlighting its core mechanisms, such as kernel approximation, Lagrangian particle transport, and boundary handling, identifying the method's grand challenges as outlined by the SPHERIC committee. These include numerical consistency, convergence, boundary condition implementation, adaptivity, and turbulence modeling. The review also underscored SPH's flexibility in dealing with multiphase and large-deformation problems, making it an attractive alternative to conventional mesh-based methods.

A major contribution to this work lies in the systematic validation of SPH performance in a wide variety of test cases. Closed and open channel flows were simulated in both two and three dimensions across different Reynolds numbers. These simulations, conducted using both the LAMMPS and DualSPHysics codes, confirmed that SPH can deliver accurate predictions of velocity and pressure fields when model parameters are appropriately chosen. The study quantified the influence of kernel function choice, particle resolution, and domain discretization on simulation quality.

Significant attention was devoted to the treatment of boundary conditions, one of SPH's most persistent challenge. The study implemented and compared multiple strategies, including ghost particles, repulsive particles, dynamic particles etc. The mDBCs approach emerged as the most effective, demonstrating improved stability and reduced artifacts near solid boundaries. This approach proved especially advantageous in geometrically complex domains, such as those involving contractions, expansions, or submerged obstacles. Also, in highly turbulent 3D flows with complex geometries, such as the 3D sluice gate-weir system presented in subsection 3.4.2, the effective tuning of the VB parameter while choosing mDBCs for solid BCs treatment is the most efficient and highly effective option. This parameter has a strong impact in several simulations and the literature regarding it is limited.

A pivotal advancement achieved in this thesis was the implementation of a Dynamic Smagorinsky turbulence model into the DualSPHysics source code. Traditional SPH approaches to turbulence, such as the SPS model, rely on fixed parameters that may not generalize across flows. The Dynamic Smagorinsky model introduced here automatically adjusts the sub-scale (in relevance to sub-grid) viscosity based on local flow characteristics, eliminating the need for manual calibration. Analysis revealed enhanced alignment with the logarithmic velocity profile near walls, better prediction of turbulence intensity, and overall improved physical realism in simulations of both open and closed channel turbulent flows.

Beyond validation and method development, this thesis also demonstrated the practical relevance of SPH through its application to a realistic 3D open channel case with submerged vegetation. This scenario, relevant to ecohydraulics and environmental design, tested the method’s ability to capture fluid–structure interactions, wake development, and turbulence modulation. The results compared favorably with both experimental and finite volume data, with relative errors remaining within 0-8% across most flow layers. This success highlights SPH’s growing maturity as a tool for modeling complex natural systems.

Despite its progress, the study also identified several current limitations of the SPH method. These include the computational cost associated with high-resolution 3D simulations, the sensitivity of the method to particle disorder near boundaries, and the need for better adaptivity mechanisms to optimize computational resources. While some of these issues were partially mitigated through techniques like shifting algorithms and improved boundary formulations, others remain open challenges for future research.

In conclusion, this thesis makes a substantial contribution to the field of computational fluid dynamics by enhancing SPH's modeling capabilities, offering practical guidelines for its use, and extending its reach into turbulent and real-world flow problems. The modifications introduced to DualSPHysics, particularly the implementation of the Dynamic Smagorinsky model, provide a meaningful step toward more reliable, efficient, and versatile SPH simulations. Also, this thesis can be used as a guide to anyone who wants to use SPH as a researcher or as a developer.

Moreover, this work serves as a foundation for further methodological innovations and interdisciplinary applications of SPH, ranging from microfluidics and hydraulic infrastructure to river restoration and urban flood modeling. It reinforces SPH’s potential not only as a research tool but also as a future-ready simulation framework for engineering practice.

6 Future work

While this study has provided valuable insights and advancements in the application and development of the SPH method, it also opened several promising directions for future research. These directions aim to address current limitations, enhance computational efficiency, expand physical modeling capabilities, and ultimately support the broader adoption of SPH in both academic and industrial contexts.

One of the most immediate and impactful extensions involves the implementation of adaptive resolution techniques. SPH simulations remain computationally expensive, particularly in three-dimensional and turbulent regimes. Adaptive resolution, where particle spacing varies spatially depending on the local flow features, could significantly reduce the computational cost without compromising accuracy. In regions with high gradients, such as boundary layers or shear zones, a finer particle resolution could be applied, while less dynamic areas could be simulated with coarser distributions. Such techniques, although already explored in limited contexts, need further development and validation within robust open-source SPH codes such as DualSPHysics. The most recent DualSPHysics solver (DualSPHysics v5.4, which was not published by the time of the research part of this work) proposed the ability of multi resolution approach yet needs to be validated and further developed in highly turbulent flows. This is the first out of many authors objectives.

Another important avenue is the integration of SPH with data-driven and machine learning approaches (Alexiadis, 2023). The SPH method involves several empirical parameters (e.g., artificial viscosity coefficients, smoothing lengths, density diffusion factors) that are often tuned manually or based on prior studies. Machine learning models could assist in automatically adjusting these parameters during simulation, based on flow evolution or real-time error estimation (Alexiadis et al., 2020). Furthermore, neural networks, especially graph neural networks or physics-informed models may offer new perspectives on improving kernel function design, enhancing convergence, or predicting particle behavior in under-resolved areas. Even the use of convolution neural networks (CNN) can enhance methods capabilities given that lower resolution simulations can use ML techniques to provide results that would demand long computational times.

Beyond SPH development, there is growing potential in hybrid numerical methods, which couple SPH with mesh-based solvers such as the Finite Element Method (FEM) or Finite Volume Method (FVM). For example, SPH could be employed in regions with high deformation or free-surface motion, while FEM could handle static solid structures or heat conduction problems more efficiently. Successful coupling of these methods could enable SPH to become part of more complex multiphysics simulations, such as those involving fluid–structure interaction, sediment transport, or thermal flow.

Another critical aspect for future development is high-performance computing (HPC) and parallelization optimization. Although DualSPHysics already supports GPU-based execution, the growing complexity of SPH models, particularly with turbulence closures, adaptive resolution, and advanced boundary conditions, demands further optimization. Efforts should be directed toward minimizing memory usage, improving neighbor search algorithms, and enhancing load balancing across parallel threads or GPU units. This would significantly improve SPH's scalability and make it more suitable for industry-scale problems or long-duration simulations.

On the validation side, while this thesis has successfully benchmarked SPH against existing literature and experimental data, the development of new experimental setups, particularly targeting turbulent, multiphase, or environmental flows, would greatly enhance the robustness of SPH assessments. Creating high-resolution, reproducible datasets that align with SPH's Lagrangian nature would not only improve validation but also foster closer collaboration between experimentalists and SPH developers.

Furthermore, the ongoing contribution of this work to the SPH framework is an important opportunity. Open-source collaboration offers a means for rapid community-wide testing, feedback, and iterative improvement. Sharing the implemented Dynamic Smagorinsky model with the developer community would enable further refinement and allow other researchers to test the model across broader problem sets. Documentation, tutorials, and integration into the official release could accelerate SPH adoption and encourage methodological consistency across studies.

Expanding the application scope of SPH remains a major ambition. The results of this thesis indicate that SPH is already capable of handling complex geometries and turbulence-dominated regimes. Future applications may include urban flood modeling, flow in porous media, debris flow, wave–structure interaction, or ecohydraulics simulations at river-reach scale. Each new application will pose unique numerical and physical challenges, but the foundational developments in this work lay the groundwork for those future efforts.

In summary, while this thesis has answered several important questions, it has also posed new ones. The SPH method is a powerful but still evolving approach to computational fluid dynamics. The path forward will be shaped by innovations in resolution adaptivity, numerical coupling, turbulence modeling, and computational efficiency, all of which promise to elevate SPH from a research-oriented method to a widely adopted engineering tool.

References

Adami, S., Hu, X. Y., & Adams, N. A. (2012). A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231, 7057–7075. https://doi.org/10.1016/j.jcp.2012.05.005

Albano, A., le Guillou, E., Danzé, A., Moulitsas, I., Sahputra, I. H., Rahmat, A., Duque-Daza, C. A., Shang, X., Ng, K. C., Ariane, M., et al. (2021). How to modify LAMMPS: From the perspective of a particle method researcher. *ChemEngineering*, 5(2), 30. https://doi.org/10.3390/chemengineering5020030

Alexiadis, A. (2023). A minimalistic approach to physics-informed machine learning using neighbour lists as physics-optimized convolutions for inverse problems involving particle systems. Journal of Computational Physics, 473, 111750.

Alexiadis, A., Simmons, M. J. H., Stamatopoulos, K., Batchelor, H. K., & Moulitsas, I. (2020). The duality between particle methods and artificial neural networks. *Scientific reports*, *10*(1), 16247.

Alfonsi, G. (2009). Reynolds-averaged Navier–Stokes equations for turbulence modeling.

Altomare, C., Crespo, A. J., Domínguez, J. M., Gómez-Gesteira, M., Suzuki, T., & Verwaest, T. (2015). Applicability of Smoothed Particle Hydrodynamics for estimation of sea wave impact on coastal structures. *Coastal Engineering*, 96, 1-12.

Altomare, C., Crespo, A. J., Rogers, B. D., Dominguez, J. M., Gironella, X., & Gómez-Gesteira, M. (2014). Numerical modeling of armour block sea breakwater with smoothed particle hydrodynamics. *Computers & Structures*, 130, 34-45.

Altomare, C., Domínguez, J. M., Crespo, A. J. C., Suzuki, T., Caceres, I., & Gómez-Gesteira, M. (2016). Hybridization of the wave propagation model SWASH and the meshfree particle method SPH for real coastal applications. *Coastal Engineering Journal*, 57(4), 1550024-1.

Altomare, C., Gironella, X., & Crespo, A. J. (2021). Simulation of random wave overtopping by a WCSPH model. *Applied Ocean Research*, 116, 102888.

Antuono, M., Colagrossi, A., Marrone, S., & Molteni, D. (2010). Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications*, 181(3), 532-549.

Basa, M., Quinlan, N. J., & Lastiwka, M. (2009). Robustness and accuracy of SPH formulations for viscous flow. *International Journal for Numerical Methods in Fluids*, 60, 1127–1148.

Blin, L., Hadjadj, A., & Vervisch, L. (2003). Large eddy simulation of turbulent flows in reversing systems. *Journal of Turbulence*, 4, 001.

Bonet, J., & Lok, T.-S. L. (1999). Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180, 97.

Boris, J. P., Grinstein, F. F., Oran, E. S., & Kolbe, R. L. (1992). New insights into large eddy simulation. *Fluid Dynamics Research*, 10(4-6), 199.

Bulian, G., & Cercos-Pita, J. L. (2018). Co-simulation of ship motions and sloshing in tanks. *Ocean Engineering*, 152, 353-376.

Canelas, R. B., Crespo, A. J., Domínguez, J. M., Ferreira, R. M., & Gómez-Gesteira, M. (2016). SPH–DCDEM model for arbitrary geometries in free surface solid–fluid flows. *Computer Physics Communications*, 202, 131-140.

Canelas, R. B., Domínguez, J. M., Crespo, A. J. C., Gómez-Gesteira, M., & Ferreira, R. M. L. (2017). Resolved simulation of a granular-fluid flow with a coupled SPH-DCDEM model. *Journal of Hydraulic Engineering*, 143(9), 06017012.

Cao, X. Y., Tao, L., Zhang, A. M., & Ming, F. R. (2021). Smoothed particle hydrodynamics (SPH) model for coupled analysis of a damaged ship with internal sloshing in beam seas. *Physics of Fluids*, 31.

Chang, C. L., & Venkatachari, B. (2019). Dynamic Smagorinsky modeled large-eddy simulations of turbulence using tetrahedral meshes. In *AIAA Scitech 2019 Forum* (p. 1647).

Chatelain, M., & Proust, S. (2021). Open-channel flows through emergent rigid vegetation: Effects of bed roughness and shallowness on the flow structure and surface waves. *Physics of Fluids*, 33(10).

Chatzoglou, E., & Liakopoulos, A. (2022). Hydraulic jump simulation via Smoothed Particle Hydrodynamics: A critical review. In *Proceedings of the 39th IAHR World Congress*, Granada, Spain (Vol. 19, p. 24).

Chatzoglou, E., & Liakopoulos, A. (2023). Flow regimes in sluice gate-weir systems: 3D SPH-based model validation. In *Proceedings of the 17th International SPHERIC Workshop*, Rhodes, Greece.

Chatzoglou, E., & Liakopoulos, A. (2025). Turbulent flow through sluice gate and weir using Smoothed Particle Hydrodynamics: Evaluation of turbulence models, boundary conditions, and 3D effects. *Water*, 17(2), 152.

Chatzoglou, E., Liakopoulos, A., & Sofos, F. (2023). Smoothed particle hydrodynamics-based study of 3D confined microflows. *Fluids*, 8, 137.

Chaudhry, M. H. (2008). *Open-channel flow* (Vol. 523). New York: Springer.

Chiron, L., Oger, G., De Leffe, M., & Le Touzé, D. (2018). Analysis and improvements of Adaptive Particle Refinement (APR) through CPU time, accuracy and robustness considerations. *Journal of Computational Physics*, 354, 552-575.

Choi, T. S., & Kim, E. S. (2023). Numerical investigation of turbulent lid-driven flow using weakly compressible smoothed particle hydrodynamics CFD code with standard and dynamic LES models. *Nuclear Engineering and Technology*, 55(9), 3367-3382.

Cipolla, K. M., Liakopoulos, A., & Rockwell, D. O. (1998). Quantitative imaging in proper orthogonal decomposition of flow past a delta wing. *AIAA Journal*, 36(7), 1247–1255.

Colagrossi, A. (2005). A meshless Lagrangian method for free-surface and interface flows with fragmentation. *These*, Universita di Roma.

Colagrossi, A., & Landrini, M. (2003). Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2), 448-475.

Crespo, A. J. C., Gomez-Gesteira, M., & Dalrymple, R. A. (2007). Boundary conditions generated by dynamic particles in SPH methods. *Computational Materials Continua*, 5, 173–184. https://doi.org/10.3970/cmc.2007.005.173

Crespo, A., Tagliafierro, B., Martínez-Estévez, I., Domínguez, J. M., Decastro, M., Gómez-Gesteira, M., & Stansby, P. (2023). On the state-of-the-art CFD simulations for wave energy converters within the open-source numerical framework of DualSPHysics. In *Proceedings of the 15th European Wave and Tidal Energy Conference*.

Curran, J. C., & Hession, W. C. (2013). Vegetative impacts on hydraulics and sediment processes across the fluvial system. *Journal of Hydrology*, 505, 364-376.

D'Ippolito, A., Calomino, F., Alfonsi, G., & Lauria, A. (2021). Flow Resistance in Open Channel Due to Vegetation at Reach Scale: A Review. *Water*, 13(2), 116. https://doi.org/10.3390/w13020116

Dalrymple, R. A., & Knio, O. (2001). SPH modeling of water waves. In *Coastal Dynamics '01* (pp. 779-787).

Dalrymple, R. A., & Rogers, B. D. (2006). Numerical modeling of water waves with the SPH method. *Coastal Engineering*, 53(2-3), 141-147.

De Courcy, J. J., Rendall, T. C., Constantin, L., Titurus, B., & Cooper, J. E. (2024). Incompressible δ-SPH via artificial compressibility. *Computer Methods in Applied Mechanics and Engineering*, 420, 116700.

De Padova, D., Mossa, M., & Sibilla, S. (2018). SPH numerical investigation of characteristics of hydraulic jumps. *Environmental Fluid Mechanics*, 18, 849-870.

Dehnen, W., & Aly, H. (2012). Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society*, 425(2), 1068-1082.

Desbrun, M., & Cani, M. P. (1996). Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation)* (pp. 61–76). Springer-Verlag.

Di Mascio, A., Antuono, M., Colagrossi, A., & Marrone, S. (2017). Smoothed particle hydrodynamics method from a large eddy simulation perspective. *Physics of Fluids*, 29.

Domínguez, J. M., Fourtakas, G., Altomare, C., Canelas, R. B., Tafuni, A., García-Feal, O., ... & Gómez-Gesteira, M. (2022). DualSPHysics: from fluid dynamics to multiphysics problems. *Computational Particle Mechanics*, 9(5), 867-895.

Dong, X., Wang, X., & Ouyang, J. (2024). An active learning SPH method for generalized Newtonian free surface flows. *Physics of Fluids*, 36(1).

Duan, G., Matsunaga, T., & Koshizuka, S. (2022). New insights into error accumulation due to biased particle distribution in semi-implicit particle methods. *Computer Methods in Applied Mechanics and Engineering*, 388.

Dunn, C., Lopez, F., & Garcia, M. (1996). Mean flow and turbulence in a laboratory channel with simulated vegetation. *Hydrosystems Laboratory, Department of Civil Engineering, University of Illinois, Urbana-Champaign*.

Eirís, A., Ramírez, L., Couceiro, I., Fernández-Fidalgo, J., París, J., & Nogueira, X. (2023). MLS-SPH-ALE: A review of meshless-FV methods and a unifying formulation for particle discretizations. *Archives of Computational Methods in Engineering*, 30(8), 4959-4981.

El Rahi, J., Martínez-Estévez, I., Almeida Reis, R., Tagliafierro, B., Domínguez, J. M., Crespo, A. J. C., Stratigaki, V., Suzuki, T., & Troch, P. (2024). Exploring wave–vegetation interaction at stem scale: Analysis of the coupled flow–structure interactions using the SPH-based DualSPHysics code and the FEA module of Chrono. *Journal of Marine Science and Engineering*, 12(7), 1120. https://doi.org/10.3390/jmse12071120

Ellero, M., & Adams, N. A. (2011). SPH simulations of flow around a periodic array of cylinders confined in a channel. *International Journal for Numerical Methods in Engineering*, 86(8), 1027-1040.

Ellero, M., & Español, P. (2017). Everything you always wanted to know about SDPD⋆ (⋆but were afraid to ask). *Applied Mathematics and Mechanics*, 39, 103–124. https://doi.org/10.1007/s10483-018-2255-6

English, A. R. (2020). *Design of Low Residue Packs by Smoothed Particle Hydrodynamics* (Doctoral dissertation). The University of Manchester, United Kingdom.

English, A., Domínguez, J. M., Vacondio, R., Crespo, A. J. C., Stansby, P. K., Lind, S. J., & Gómez-Gesteira, M. (2019). Correction for Dynamic Boundary Conditions. In *Proceedings of the 14th International SPHERIC Workshop*, Exeter, United Kingdom.

English, A., Domínguez, J. M., Vacondio, R., Crespo, A. J. C., Stansby, P. K., Lind, S. J., & Gómez-Gesteira, M. (2022). Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): Application to tank sloshing, dam break, and fish pass problems. *Computational Particle Mechanics*, 9(5), 1-15.

Español, P. (1995). Hydrodynamics from dissipative particle dynamics. *Physical Review E*, 52, 1734.

Eymard, R., Gallouët, T., & Herbin, R. (2000). Finite volume methods. In *Handbook of Numerical Analysis* (Vol. 7, pp. 713-1018).

Feng, R., Fourtakas, G., Rogers, B. D., & Lombardi, D. (2024). A general smoothed particle hydrodynamics (SPH) formulation for coupled liquid flow and solid deformation in porous media. *Computer Methods in Applied Mechanics and Engineering*, 419, 116581.

Fourey, G., Hermange, C., Le Touzé, D., & Oger, G. (2017). An efficient FSI coupling strategy between smoothed particle hydrodynamics and finite element methods. *Computer Physics Communications*, 217, 66-81.

Fourtakas, G., Dominguez, J. M., Vacondio, R., & Rogers, B. D. (2019). Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Computers & Fluids*, 190, 346-361.

Fourtakas, G., Vacondio, R., Domínguez, J. M., & Rogers, B. D. (2020). Improved density diffusion term for long duration wave propagation. In *Proceedings of the International SPHERIC Workshop*, Harbin, China.

Frontiere, N., Raskin, C. D., & Owen, J. M. (2017). CRKSPH–a conservative reproducing kernel smoothed particle hydrodynamics scheme. *Journal of Computational Physics*, 332, 160-209.

Ganzenmüller, G. C., Steinhauser, M. O., Van Liedekerke, P., & Leuven, K. U. (2011). The implementation of smooth particle hydrodynamics in LAMMPS. *Paul Van Liedekerke Katholieke Universiteit Leuven*, 1(1-26), 31. Retrieved from http://lammps.sandia.gov/doc/PDF/SPH_LAMMPS_userguide.pdf

Gao, T., Qiu, H., & Fu, L. (2022). A block-based adaptive particle refinement SPH method for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 399, 115356.

Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3, 1760–1765.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181, 375–389.

Gómez-Gesteira, M., Crespo, A. J., Rogers, B. D., Dalrymple, R. A., Dominguez, J. M., & Barreiro, A. (2012). SPHysics–development of a free-surface fluid solver–Part 2: Efficiency and test cases. *Computers & Geosciences*, 48, 300-307.

Gotoh, H., & Khayyer, A. (2016). Current achievements and future perspectives for projection-based particle methods with applications in ocean engineering. *Journal of Ocean Engineering and Marine Energy*, 2(3), 251–278. https://doi.org/10.1007/s40722-016-0049-3

Gotoh, H., Ikari, H., Memita, T., & Sakai, T. (2005). Lagrangian particle method for simulation of wave overtopping on a vertical seawall. *Coastal Engineering Journal*, 47, 157–181.

Gray, J. P., Monaghan, J. J., & Swift, R. P. (2001). SPH elastic dynamics. *Computer Methods in Applied Mechanics and Engineering*, 190(49-50), 6641-6662.

Henry, H. R. (1950). Discussion of Diffusion of submerged jets. In *ASCE Transactions* (Vol. 115, pp. 687-694).

Hérault, A., Bilotta, G., & Dalrymple, R. A. (2010). SPH on GPU with CUDA. *Journal of Hydraulic Research*, 48(S1), 74–79. https://doi.org/10.1080/00221686.2010.9641247

Hopkins, P. F. (2015). A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society*, 450(1), 53-110.

Hu, X. Y., & Adams, N. A. (2006). A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(2), 844-861.

Huang, C., Lei, J. M., Liu, M. B., & Peng, X. Y. (2016). An improved KGF-SPH with a novel discrete scheme of Laplacian operator for viscous incompressible fluid flows. *International Journal for Numerical Methods in Fluids*, 81, 377.

Huebner, K. H., Dewhirst, D. L., Smith, D. E., & Byrom, T. G. (2001). *The finite element method for engineers*. John Wiley & Sons.

Husain, S. M., Muhammed, J. R., Karunarathna, H. U., & Reeve, D. E. (2014). Investigation of pressure variations over stepped spillways using smooth particle hydrodynamics. *Advances in Water Resources*, 66, 52-69.

Jin, C., & Zhang, J. (2022). Numerical investigation of the wave interaction with flexible vegetation: Model setup and validation for a single stem study case. *Anthropocene Coasts*, 5(3). https://doi.org/10.1007/s44218-022-00003-5.

Kasiteropoulou, D., Liakopoulos, A., Michalolias, N., & Keramaris, E. (2017). Numerical modeling and analysis of turbulent flow in an open channel with submerged vegetation. *Environmental Processes*, 4, 47-61.

Kazemi, E., Koll, K., Tait, S., & Shao, S. (2020). SPH modeling of turbulent open channel flow over and within natural gravel beds with rough interfacial boundaries. *Advances in Water Resources*, 140, 103557.

Kazemi, E., Nichols, A., Tait, S., & Shao, S. (2017). SPH modeling of depth-limited turbulent open channel flows over rough boundaries. *International Journal for Numerical Methods in Fluids*, 83(1), 3-27.

Khani, S., & Waite, M. L. (2015). Large eddy simulations of stratified turbulence: The dynamic Smagorinsky model. *Journal of Fluid Mechanics*, 773, 327–344.

Korzani, M. G., Galindo-Torres, S. A., Scheuermann, A., & Williams, D. J. (2017). Parametric study on smoothed particle hydrodynamics for accurate determination of drag coefficient for a circular cylinder. *Water Science and Engineering*, 10(2), 143-153.

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggen, E. M. (2017). *The lattice Boltzmann method*. Springer International Publishing.

Kulasegaram, S., Bonet, J., Lewis, R. W., & Profit, M. (2004). A variational formulation-based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics*, 33, 316–325. https://doi.org/10.1007/s00466-003-0534-0

Ladický, L. U., Jeong, S., Solenthaler, B., Pollefeys, M., & Gross, M. (2015). Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics (TOG)*, 34(6), 1-9.

Lee, E.-S., Moulinec, C., Xu, R., Violeau, D., Laurence, D., & Stansby, P. (2008). Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh-free particle method. *Journal of Computational Physics*, 227(18), 8417-8436.

Lee, M., & Moser, R. D. (2015). Direct numerical simulation of turbulent channel flow up to. *Journal of Fluid Mechanics*, 774, 395-415.

Leimkuhler, B., Matthews, C., & Stoltz, G. (2016). The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. *IMA Journal of Numerical Analysis*, 36(1), 13-79.

Leroy, A., Violeau, D., Ferrand, M., & Kassiotis, C. (2014). Unified semianalytical wall boundary conditions applied to 2-D incompressible SPH. *Journal of Computational Physics*, 261, 106–129. https://doi.org/10.1016/j.jcp.2013.12.035

Lesieur, M., Métais, O., & Comte, P. (2005). *Large-eddy simulations of turbulence*. Cambridge University Press.

Li, D., Peng, Z., Liu, G., & Wei, C. (2023). Flow structures in open channels with emergent rigid vegetation: A review. *Water*, 15(23), 4121. https://doi.org/10.3390/w15234121

Liakopoulos, A. (1985). Computation of high-speed turbulent boundary-layer flows using the k–ε turbulence model. *International Journal for Numerical Methods in Fluids*, 5, 81–97.

Liakopoulos, A. (2019). *Fluid Mechanics* (2nd ed.). Tziolas Publications. Athina, Greece.

Liakopoulos, A. (2020). *Hydraulics* (3rd ed.). Tziolas Publications. Thessaloniki, Greece.

Liakopoulos, A., & Palasis, A. (2024). Turbulent channel flow: Direct numerical simulation-data-driven modeling. *Fluids*, 9, 62.

Liakopoulos, A., Blythe, P. A., & Gunes, H. (1997). A reduced dynamical model of convective flows in tall laterally heated cavities. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 453(1958), 663–672.

Liakopoulos, A., Sofos, F., & Karakasidis, T. E. (2017). Darcy-Weisbach friction factor at the nanoscale: From atomistic calculations to continuum models. *Physics of Fluids*, 29, 052003. https://doi.org/10.1063/1.4982667.

Libersky, L. (1991). Smooth particle hydrodynamics with strength of materials. In *Lecture Notes in Physics* (Vol. 395, pp. 248-257).

Lind, S. J., Stansby, P. K., Rogers, B. D., & Lloyd, P. M. (2015). Numerical predictions of water–air wave slam using incompressible–compressible smoothed particle hydrodynamics. *Applied Ocean Research*, 49, 57-71.

Lind, S. J., Xu, R., Stansby, P. K., & Rogers, B. D. (2012). Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4), 1499-1523.

Liu, G. R. (2003). *Mesh Free methods: Moving beyond the finite element method*. CRC Press.

Liu, M. B., & Liu, G. R. (2006). Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, 56, 19-36.

Liu, M. B., Liu, G. R., & Lam, K. Y. (2003). Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 155(2), 263-284.

Lo, E. Y., & Shao, S. (2002). Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Applied Ocean Research*, 24(5), 275-286.

Lorenz, S., Pusch, M. T., & Blaschke, U. (2015). Minimum shoreline restoration requirements to improve the ecological status of a north-eastern German glacial lowland lake in an urban landscape. *Fundamental and Applied Limnology*, 186, 323–332.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82(12), 1013-1024.

Manenti, S., Sibilla, S., Gallati, M., Agate, G., & Guandalini, R. (2012). SPH simulation of sediment flushing induced by a rapid water flow. *Journal of Hydraulic Engineering*, 138(3), 272-284.

Marrone, S., Antuono, M. A. G. D., Colagrossi, A., Colicchio, G., Le Touzé, D., & Graziani, G. (2011). δ-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16), 1526-1542.

Marrone, S., Saltari, F., Michel, J., & Mastroddi, F. (2023). SPH modeling of dissipative sloshing flows under violent vertical harmonic excitation. *Journal of Fluids and Structures*, 119, 103877.

Martínez-Estévez, I., Domínguez, J. M., Tagliafierro, B., Canelas, R. B., García-Feal, O., Crespo, A. J., & Gómez-Gesteira, M. (2023). Coupling of an SPH-based solver with a multiphysics library. *Computer Physics Communications*, 283, 108581.

Martínez-Estévez, I., Tagliafierro, B., El Rahi, J., Domínguez, J. M., Crespo, A. J., Troch, P., & Gómez-Gesteira, M. (2023). Coupling an SPH-based solver with an FEA structural solver to simulate free surface flows interacting with flexible structures. *Computer Methods in Applied Mechanics and Engineering*, 410, 115989.

Mayrhofer, A., Laurence, D., Rogers, B. D., & Violeau, D. (2015). DNS and LES of 3-D wall-bounded turbulence using smoothed particle hydrodynamics. *Computers & Fluids*, 115, 86–107.

Mayrhofer, A., Rogers, B. D., Violeau, D., & Ferrand, M. (2013). Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Computer Physics Communications*, 184, 2515–2527.

Meneveau, C., Lund, T. S., & Cabot, W. H. (1996). A Lagrangian dynamic subgrid-scale model of turbulence. *Journal of Fluid Mechanics*, 319, 353–385.

Meringolo, D. (2018). Numerical simulation of granular collapses using SPH. *Granular Matter*, 20, 45.

Meyers, J., & Sagaut, P. (2006). On the model coefficients for the standard and the variational multi-scale Smagorinsky model. *Journal of Fluid Mechanics*, 569, 287–319.

Min, L., Su, X., Kazemi, E., Jin, X., & Khayyer, A. (2025). Review of smoothed particle hydrodynamics modeling of fluid flows in porous media with a focus on hydraulic, coastal, and ocean engineering applications. *Physics of Fluids*, 37(2), 021303. https://doi.org/10.1063/5.0252125

Moin, P., & Mahesh, K. (1998). Direct numerical simulation: A tool in turbulence research. *Annual Review of Fluid Mechanics*, 30(1), 539-578.

Molteni, D., & Colagrossi, A. (2009). A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH. *Computer Physics Communications*, 180, 861–872.

Monaghan, J. J. (1985). Extrapolating B splines for interpolation. *Journal of Computational Physics*, 60(2), 253–262.

Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30, 543–574.

Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, 110, 399–406. https://doi.org/10.1006/jcph.1994.1034

Monaghan, J. J. (2002). SPH compressible turbulence. *Monthly Notices of the Royal Astronomical Society*, 335, 843–852.

Monaghan, J. J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8), 1703–1759.

Monaghan, J. J., & Gingold, R. A. (1983). Shock simulation by the Particle Method SPH. *Journal of Computational Physics*, 52, 374–389. https://doi.org/10.1016/0021-9991(90)9036-0

Monaghan, J. J., & Kajtar, J. B. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10), 1811-1820. https://doi.org/10.1016/j.cpc.2009.05.008

Monaghan, J. J., & Kos, A. (1999). Solitary waves on a Cretan beach. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 125(3), 145-155.

Monaghan, J. J., & Lattanzio, J. C. (1985). A refined method for astrophysical problems. *Astronomy and Astrophysics*, 149, 135–143.

Monaghan, J. J., & Pongracic, H. (1985). Artificial viscosity for particle methods. *Applied Numerical Mathematics*, 1(3), 187-194.

Monaghan, J. J., Kos, A., & Issa, N. (2003). Fluid motion generated by impact. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 129(6), 250-259.

Morris, J. P., Fox, P. J., & Zhu, Y. (1997). Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1), 214-226.

Morris, J. S., Frith, C. D., Perrett, D. I., Rowland, D., Young, A. W., Calder, A. J., & Dolan, R. J. (1996). A differential neural response in the human amygdala to fearful and happy facial expressions. *Nature*, 383(6603), 812-815.

Mossa, M., & De Padova, D. (2025). Interaction between waves and vegetation. *Scientific Reports*, 15, 6157. https://doi.org/10.1038/s41598-025-89627-4

Müller, M., Charypar, D., & Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (pp. 154-159).

Murante, G., Borgani, S., Brunino, R., & Cha, S. H. (2011). Hydrodynamic simulations with the Godunov smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 417(1), 136-153.

Muta, A., & Ramachandran, P. (2022). Efficient and accurate adaptive resolution for weakly-compressible SPH. *Computer Methods in Applied Mechanics and Engineering*, 395, 115019.

Nasar, A. M. A., Fourtakas, G., Lind, S. J., Rogers, B. D., Stansby, P. K., & King, J. R. C. (2021). High-order velocity and pressure wall boundary conditions in Eulerian Incompressible SPH. *Journal of Computational Physics*, 434, 109793. https://doi.org/10.1016/j.jcp.2020.109793

Nepf, H., & Ghisalberti, M. (2008). Flow and transport in channels with submerged vegetation. *Acta Geophysica*, 56, 753-777.

Ni, X., Feng, W. B., & Wu, D. (2014). Numerical simulations of wave interactions with vertical wave barriers using the SPH method. *International Journal for Numerical Methods in Fluids*, 76(4), 223-245.

Novak, G., Domínguez, J. M., Tafuni, A., Silva, A. T., Pengal, P., Četina, M., & Žagar, D. (2021). 3-D numerical study of a bottom ramp fish passage using smoothed particle hydrodynamics. *Water*, 13(11), 1595.

O'Hare, M. T. (2015). Aquatic vegetation—A primer for hydrodynamic specialists. *Journal of Hydraulic Research*, 53, 687–696.

Oger, G., Marrone, S., Le Touzé, D., & De Leffe, M. (2016). SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms. *Journal of Computational Physics*, 313, 76–98.

Park, N., & Mahesh, K. (2009). Reduction of the Germano-identity error in the dynamic Smagorinsky model. *Physics of Fluids*, 21.

Parshikov, A. N., Medin, S. A., Loukashenko, I. I., & Milekhin, V. A. (2000). Improvements in SPH method by means of interparticle contact algorithm and analysis of perforation tests at moderate projectile velocities. *International Journal of Impact Engineering*, 24(8), 779-796.

Perdikaris, P., Grinberg, L., & Karniadakis, G. E. (2016). Multiscale modeling and simulation of Brain Blood Flow. *Physics of Fluids*, 28, 021304. https://doi.org/10.1063/1.4941315

Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3), 220-252.

Pilloton, C., Sun, P. N., Zhang, X., & Colagrossi, A. (2024). Volume conservation issue within SPH models for long-time simulations of violent free-surface flows. *Computer Methods in Applied Mechanics and Engineering*, 419, 116640.

Pivkin, I. V., Caswell, B., & Karniadakis, G. E. (2010). Dissipative Particle Dynamics. *Reviews in Computational Chemistry*, 27, 85–110. https://doi.org/10.1002/9780470890905.ch2.

Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117, 1–19. https://doi.org/10.1006/jcph.1995.1039

Price, D. J., Wurster, J., Tricco, T. S., Nixon, C. J., Toupin, S., Pettitt, A. R., ... & Lodato, G. (2018). Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics code for astrophysics. *Publications of the Astronomical Society of Australia*, 35, e031. https://doi.org/10.1017/pasa.2018.25

Qiu, H., Li, P., Wang, W., Shi, P., Zhang, J., & Chen, H. (2024). Impacts of emergent rigid vegetation patches on flow characteristics of open channels. *Journal of Hydroinformatics*, 26(7), 1629-1647.

Qu, J., & Yu, J. (2020). A numerical modeling of flows in an open channel with emergent vegetation. *Journal of Hydraulic Research*, 59(2), 250–262.https://doi.org/10.1080/00221686.2020.1770877

Ramachandran, P., Bhosale, A., Puri, K., Negi, P., Muta, A., Dinesh, A., ... & Agarwal, A. (2021). PySPH: A Python-based framework for smoothed particle hydrodynamics. *ACM Transactions on Mathematical Software (TOMS)*, 47(4), 1-38.

Randles, P. W., & Libersky, L. D. (1996). Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4), 375-408.

Rao, L., Wang, P. F., Dai, Q. S., & Wang, C. (2018). The coupling between hydrodynamic and purification efficiencies of ecological porous spur-dike in field drainage ditch. *Journal of Hydrodynamics*, 30, 373–383.

Reece, S., Clarke, D., & Lowe, T. (2024). High-resolution SPH simulation of tsunami wave impact. *Ocean Engineering*, 281, 114351.

Ricci, F., Vacondio, R., & Tafuni, A. (2023). Direct numerical simulation of three-dimensional isotropic turbulence with smoothed particle hydrodynamics. *Physics of Fluids*, 35(6).

Robinson, M., & Monaghan, J. J. (2012). Direct numerical simulation of decaying two-dimensional turbulence in a no-slip square box using smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 70(1), 37-55.

Roth, A., & Hager, W. H. (1999). Underflow of standard sluice gate. *Experimental Fluids*, 27, 339–350.

Roubtsova, V., & Kahawita, R. (2006). The SPH technique applied to free surface flows. *Computers & Fluids*, 35(10), 1359-1371.

Sahan, R. A., Koc-Sahan, N., Albin, D. C., & Liakopoulos, A. (1997). Artificial neural network-based modeling and intelligent control of transitional flows. In *Proceedings of the 1997 IEEE International Conference on Control Applications* (pp. 359–364).

Schaller, M., Gonnet, P., Chalk, A. B., & Draper, P. W. (2016, June). SWIFT: Using task-based parallelism, fully asynchronous communication, and graph partition-based domain decomposition for strong scaling on more than 100,000 cores. In *Proceedings of the Platform for Advanced Scientific Computing Conference* (pp. 1-10).

Serván-Camas, B., Cercós-Pita, J. L., Colom-Cobb, J., García-Espinosa, J., & Souto-Iglesias, A. (2016). Time domain simulation of coupled sloshing–seakeeping problems by SPH–FEM coupling. *Ocean Engineering*, 123, 383-396.

Shadloo, M., Oger, G., & Le Touzé, D. (2016). Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Computers & Fluids*, 136, 11–34. https://doi.org/10.1016/j.compfluid.2016.05.029

Shao, J. R., Li, H. Q., Liu, G. R., & Liu, M. B. (2012). An improved SPH method for modeling liquid sloshing dynamics. *Computers & Structures*, 100, 18-26.

Shen, J., Tang, T., & Wang, L. L. (2011). *Spectral methods: algorithms, analysis, and applications* (Vol. 41). Springer Science & Business Media.

Sibil, R. (2024). Evaluation of turbulence models for the prediction of flow properties in vegetated channels. *Physics of Fluids*, 36(9), 095158.

Sibilla, S. (2015). An algorithm to improve consistency in smoothed particle hydrodynamics. *Computers & Fluids*, 118, 148-158.

Skillen, A., Lind, S., Stansby, P. K., & Rogers, B. D. (2013). Incompressible Smoothed Particle Hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body-water slam and efficient wave-body interaction. *Computer Methods in Applied Mechanics and Engineering*, 256, 163-173.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91, 99–114.

Sofos, F., Chatzoglou, E., & Liakopoulos, A. (2022). An assessment of SPH simulations of sudden expansion/contraction 3-D channel flows. *Computational Particle Mechanics*, 1-15.

Sofos, F., Karakasidis, T. E., & Liakopoulos, A. (2013). Fluid flow at the nanoscale: How fluid properties deviate from the bulk. *Nanoscale and Nanotechnology Letters*, 5, 457–460. https://doi.org/10.1166/nnl.2013.1555.

Sofos, F., Sofiadis, G., Chatzoglou, E., Palasis, A., Karakasidis, T. E., & Liakopoulos, A. (2024). From sparse to dense representations in open channel flow images with convolutional neural networks. *Inventions*, *9*(2), 27. https://doi.org/10.3390/inventions9020027.

Springel, V. (2005). The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364(4), 1105–1134. https://doi.org/10.1111/j.1365-2966.2005.09655.x

Springel, V. (2010). E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 401(2), 791-851.

Sun, P. N., Colagrossi, A., Marrone, S., & Zhang, A. M. (2017). The δplus-SPH model: Simple procedures for a further improvement of the SPH scheme. *Computer Methods in Applied Mechanics and Engineering*, 315, 25-49.

Sun, P. N., Colagrossi, A., Marrone, S., Antuono, M., & Zhang, A. M. (2019). A consistent approach to particle shifting in the δ-Plus-SPH model. *Computer Methods in Applied Mechanics and Engineering*, 348, 912-934.

Sun, P., Zhang, A. M., Marrone, S., & Ming, F. (2018). An accurate and efficient SPH modeling of the water entry of circular cylinders. *Applied Ocean Research*, 72, 60-75.

Swamee, P. K. (1992). Sluice-gate discharge equations. *Journal of Irrigation and Drainage Engineering*, 118(1), 56-60.

Swegle, J. W., Hicks, D. L., & Attaway, S. W. (1995). Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116(1), 123-134.

Tafuni, A., Domínguez, J. M., Vacondio, R., & Crespo, A. J. C. (2018). A versatile algorithm for the treatment of open boundary conditions in Smoothed Particle Hydrodynamics GPU models. *Computer Methods in Applied Mechanics and Engineering*, 342, 604-624.

Tafuni, A., Sahin, I., & Hyman, M. (2016). Numerical investigation of wave elevation and bottom pressure generated by a planing hull in finite-depth water. *Applied Ocean Research*, 58, 281-291.

Tamai, T., & Koshizuka, S. (2014). Least squares moving particle semi-implicit method: An arbitrary high order accurate meshfree Lagrangian approach for incompressible flow with free surfaces. *Computational Particle Mechanics*, 1, 277-305.

Thompson, P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., & Plimpton, S. J. (2022). LAMMPS - A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271, 10817.

Toshev, A. P., Erbesdobler, J. A., Adams, N. A., & Brandstetter, J. (2024). Neural SPH: Improved neural modeling of Lagrangian fluid dynamics. *arXiv preprint arXiv:2402.06275*.

Vacondio, R., Altomare, C., De Leffe, M., et al. (2021). Grand challenges for Smoothed Particle Hydrodynamics numerical schemes. *Computational Particle Mechanics*, 8, 575–588. https://doi.org/10.1007/s40571-020-00354-1.

Vacondio, R., Rogers, B. D., Stansby, P. K., Mignosa, P., & Feldman, J. (2013). Variable resolution for SPH: A dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering*, 256, 132-148.

Verbrugghe, T., Domínguez, J. M., Crespo, A. J., Altomare, C., Stratigaki, V., Troch, P., & Kortenhaus, A. (2018). Coupling methodology for smoothed particle hydrodynamics modeling of non-linear wave-structure interactions. *Coastal Engineering*, 138, 184-198.

Verlet, L. (1967). Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, 159(1), 98.

Versteeg, H. K., & Malalasekera, W. (1995). *The finite volume method: An introduction to computational fluid dynamics*. Pearson Education.

Vila, J. P. (1999). On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*, 9(02), 161-209.

Violeau, D. (2012). *Fluid mechanics and the SPH method: theory and applications*. Oxford University Press.

Violeau, D., & Issa, R. (2007). Numerical modeling of complex turbulent free-surface flows with the SPH method: An overview. *International Journal for Numerical Methods in Fluids*, 53(2), 277-304.

Violeau, D., & Rogers, B. D. (2016). Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future. *Journal of Hydraulic Research*, 54(1), 1-26.

Wang, J., Hu, W., Zhang, X., & Pan, W. (2019). Modeling heat transfer subject to inhomogeneous Neumann boundary conditions by smoothed particle hydrodynamics and peridynamics. *International Journal of Heat and Mass Transfer*, 139, 948–962. https://doi.org/10.1016/j.ijheatmasstransfer.2019.05.054

Wang, Z. B., Chen, R., Wang, H., Liao, Q., Zhu, X., & Li, S. Z. (2016). An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modeling*, 40(23), 9625–9655. https://doi.org/10.1016/j.apm.2016.06.030

Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4, 389-396.

Wendland, H. (2004). *Scattered Data Approximation*. Cambridge University Press.

White, F. M. (2006). *Viscous Fluid Flow*. McGraw-Hill: New York, NY, USA.

Woodward, M. J., Tian, Y., Hyett, C., Fryer, C., Livescu, D., Stepanov, M., & Chertkov, M. (2021). Physics informed machine learning of SPH: Machine learning Lagrangian turbulence.

Xu, R., Stansby, P., & Laurence, D. (2009). Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*, 228(18), 6703-6725.

Xu, Y., Yang, G., Zhu, Y., & Hu, D. (2021). A coupled SPH–FVM method for simulating incompressible interfacial flows with large density difference. *Engineering Analysis with Boundary Elements*, 128, 227-243.

Yang, X., Kong, S. C., Liu, M., & Liu, Q. (2021). Smoothed particle hydrodynamics with adaptive spatial resolution (SPH-ASR) for free surface flows. *Journal of Computational Physics*, 443, 110539.

Yang, X., Liang, G., Zhang, G., Zhang, Z., & Sun, Z. (2024). A coupled SPH-SPIM solver for fluid-structure interaction with nonlinear deformation. *Computer Methods in Applied Mechanics and Engineering*, 427, 117015.

Ye, T., Pan, D., Huang, C., & Liu, M. (2019). Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications. *Physics of Fluids*, 31(1), 011301. https://doi.org/10.1063/1.5068697

Yoosefdoost, M., & Lubitz, W. D. (2022). Sluice gate design and calibration: Simplified models to distinguish flow conditions and estimate discharge coefficient and flow rate. *Water*, 14(8), 1215.

Zeng, C., Bai, Y., Zhou, J., Qiu, F., Ding, S., Hu, Y., & Wang, L. (2022). Large Eddy Simulation of Compound Open Channel Flows with Floodplain Vegetation. *Water*, 14(23), 3951. https://doi.org/10.3390/w14233951

Zhang, A., Sun, P., & Ming, F. (2015). An SPH modeling of bubble rising and coalescing in three dimensions. *Computer Methods in Applied Mechanics and Engineering*, 294, 189-209.

Zhang, C., Rezavand, M., Zhu, Y., Yu, Y., Wu, D., Zhang, W., ... & Hu, X. (2021). SPHinXsys: An open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics. *Computer Physics Communications*, 267, 108066.

Zhang, G., Yang, X., Liang, G., Zheng, K., & Zhang, Z. (2024). Numerical simulation of sloshing flows with elastic structure by coupling δ+-SPH and SPIM. *Engineering Analysis with Boundary Elements*, 165, 105764.

Zhang, Q., Liu, X., Dong, X., Yin, L., & Cheng, Z. (2024). A volume-adaptive mesh-free model for FSI simulation of cavitation erosion with bubble collapse. *Computational Particle Mechanics*, 11(5), 2325-2351.

Zhang, Z. L., Walayat, K., Chang, J. Z., & Liu, M. B. (2018). Meshfree modeling of a fluid-particle two-phase flow with an improved SPH method. *International Journal for Numerical Methods in Engineering*, 116, 530–569. https://doi.org/10.1002/nme.5935

Zhou, J., Fu, B., Gao, G., Lü, Y., Liu, Y., Lü, N., & Wang, S. (2016). Effects of precipitation and restoration vegetation on soil erosion in a semi-arid environment in the Loess Plateau, China. *Catena*, 137, 1-11.

Appendix

A

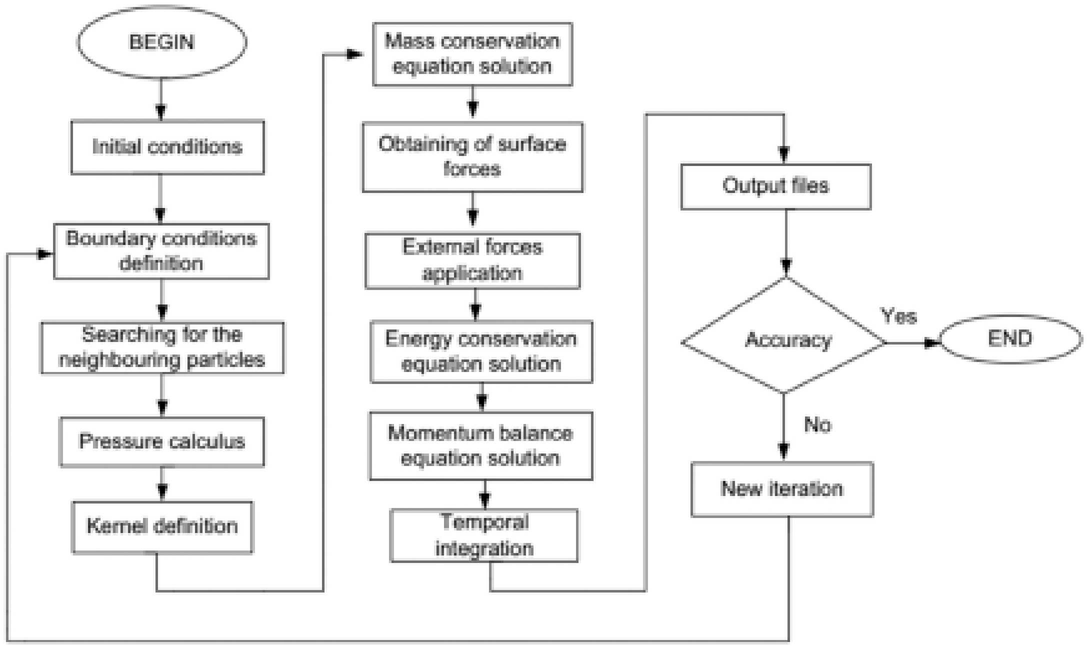


Figure A1. LAMMPS-SPH code case execution flow chart.

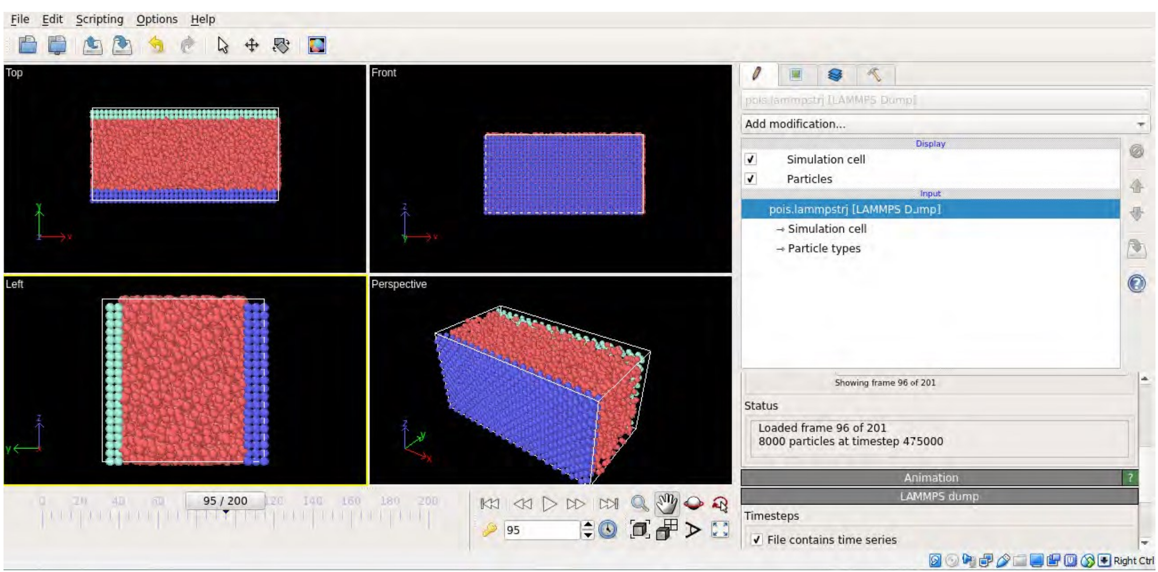


Figure A2. Snapshot of OVITO visualization tool.

B

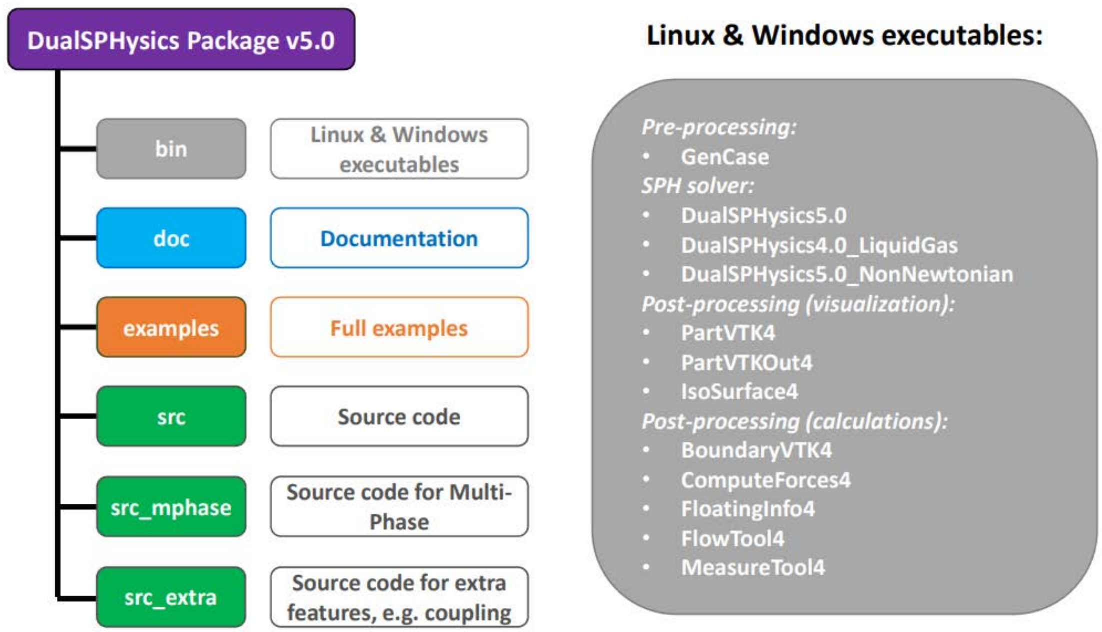


Figure B1. DualSPHysics project v5.0-Download.

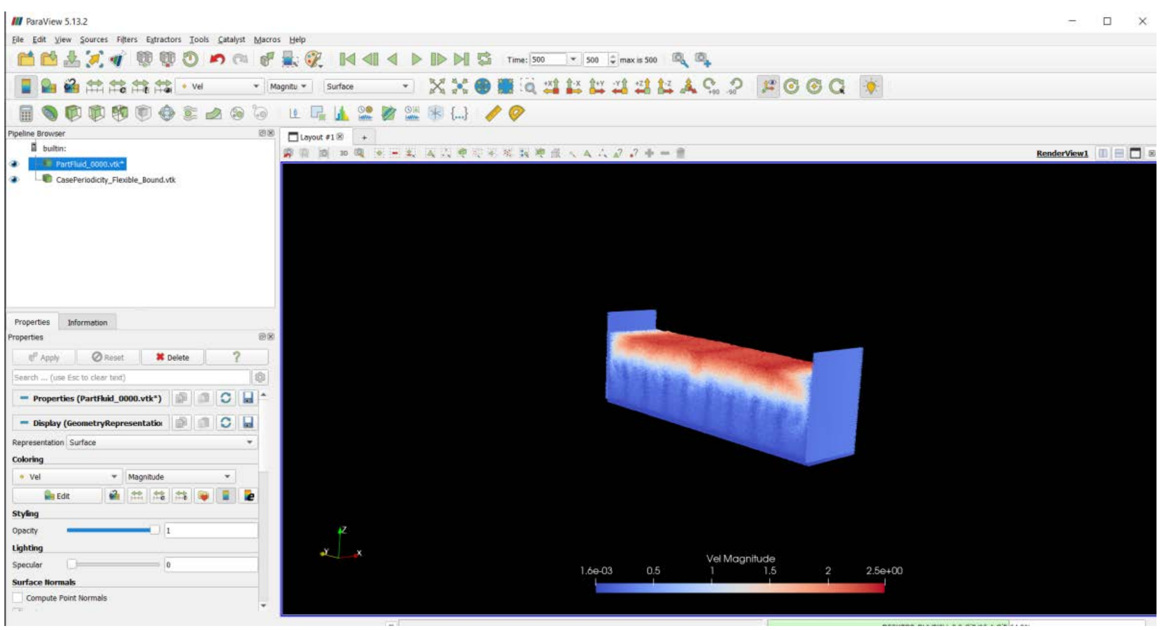


Figure B2. Paraview visualization tool.

C

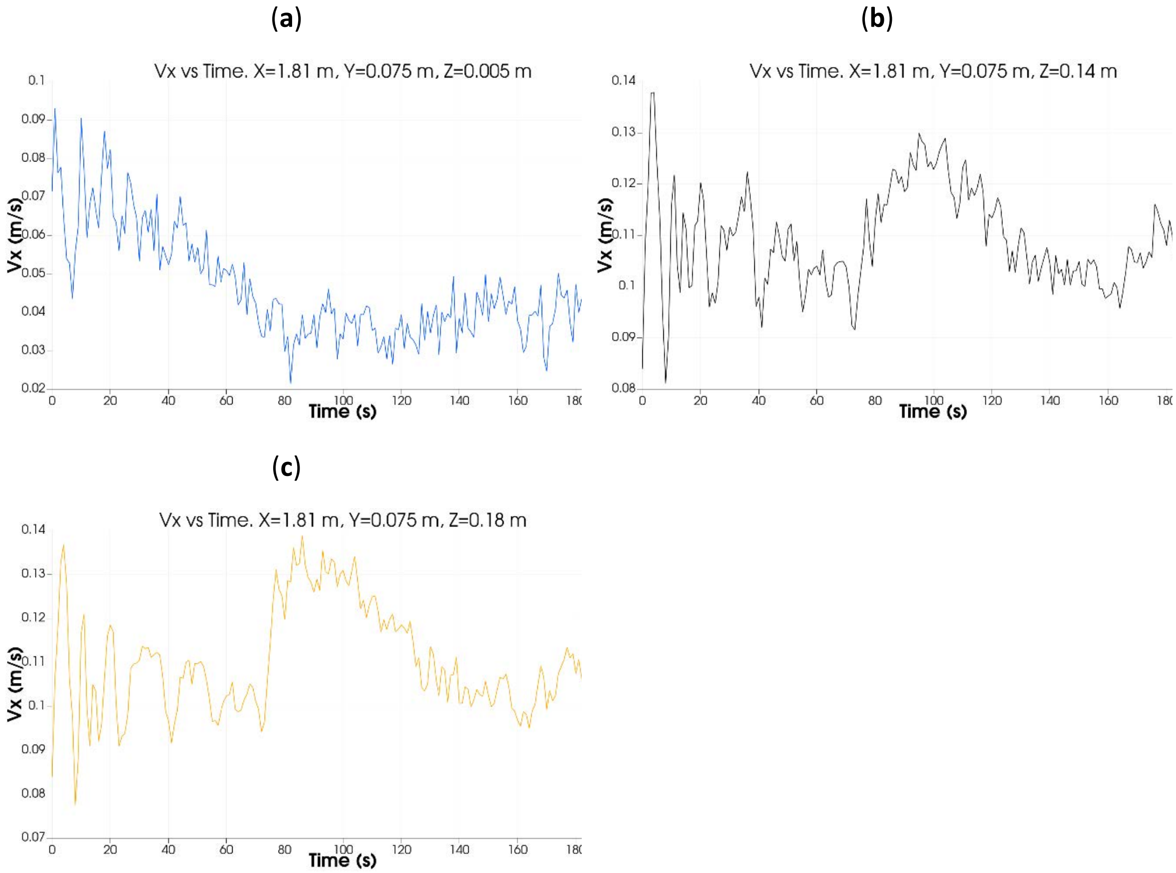


Figure C1. AVM. The time series of the X-component of velocity at X = 1.81 m (upstream of the sluice gate). (**a**) Point X = 1.81 m, Y = 0.075 m, Z = 0.005 m; (**b**) Point X = 1.81 m, Y = 0.075 m, Z = 0.14 m; (**c**) Point X = 1.81 m, Y = 0.075 m, Z = 0.18 m.

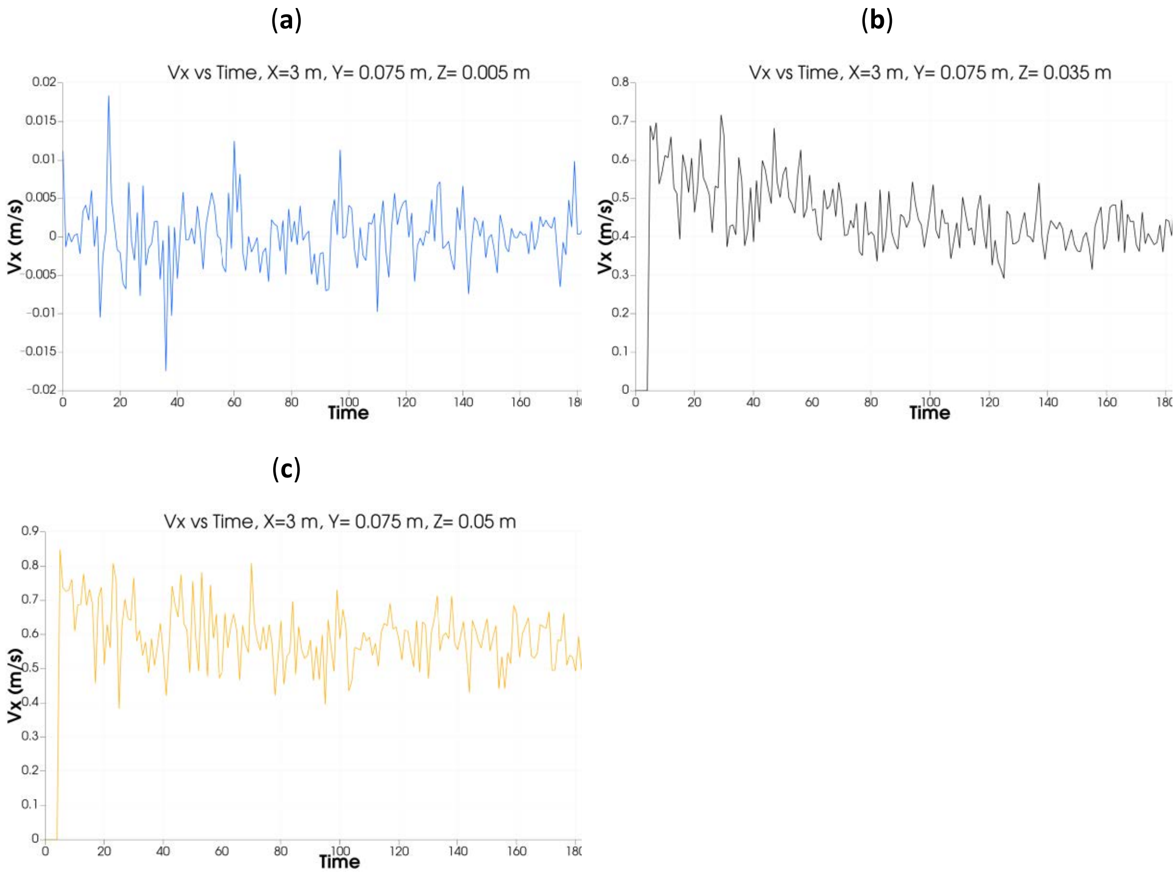


Figure C2. AVM. The time series of the X-component of velocity at X = 3 m (downstream of the hydraulic jump). (**a**) Point X = 3 m, Y = 0.075 m, Z = 0.005 m; (**b**) Point X = 3 m, Y = 0.075 m, Z = 0.035 m; (**c**) Point X = 3 m, Y = 0.075 m, Z = 0.05 m.

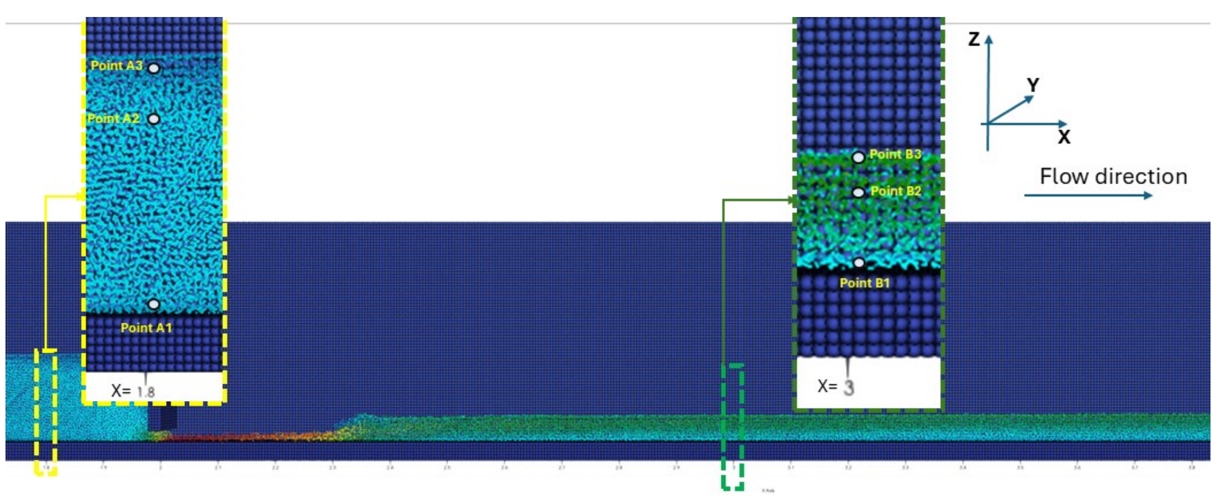


Figure C3. The visualization of L-SPS at Y = 0.075 m, showing flow characteristics on a clipped plane with projections on the channel's back wall. The locations of measurement points (A1–A3, B1–B3) at X = 1.8 m and X = 3 m are highlighted.

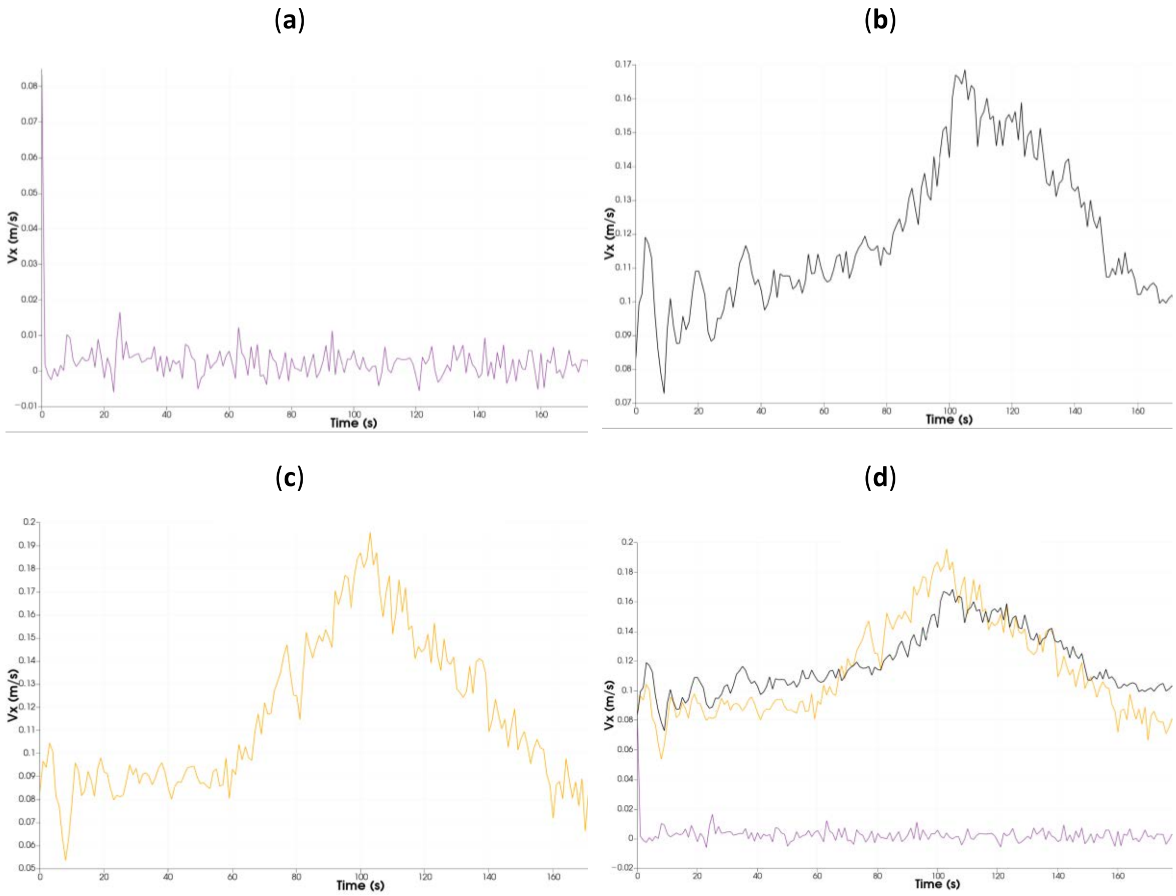


Figure C4. L-SPS. The time series of the X-component of velocity at X = 1.81 m (upstream of the sluice gate). (**a**) Point A1, X = 1.81 m, Y = 0.075 m, Z = 0.005 m; (**b**) Point A2, X = 1.81 m, Y = 0.075 m, Z = 0.14 m; (**c**) Point A3, X = 1.81 m, Y = 0.075 m, Z = 0.18 m; (**d**) all points.

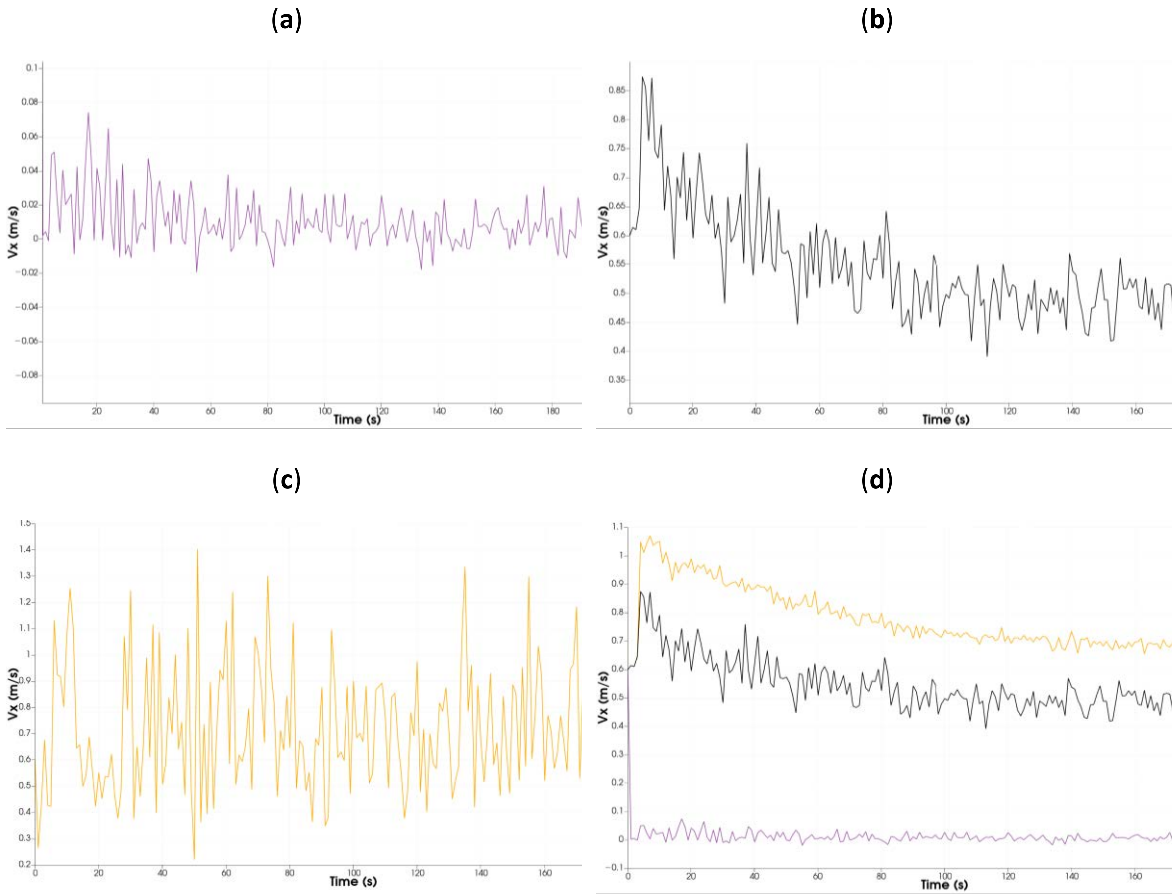


Figure C5. L-SPS. The time series of the X-component of velocity at X = 3 m (downstream of the hydraulic jump). (**a**) Point B1, X = 3 m, Y = 0.075 m, Z = 0.005 m; (**b**) Point B2, X = 3 m, Y = 0.075 m, Z = 0.035 m; (**c**) Point B3, X = 3 m, Y = 0.075 m, Z = 0.05 m; (**d**) all points.

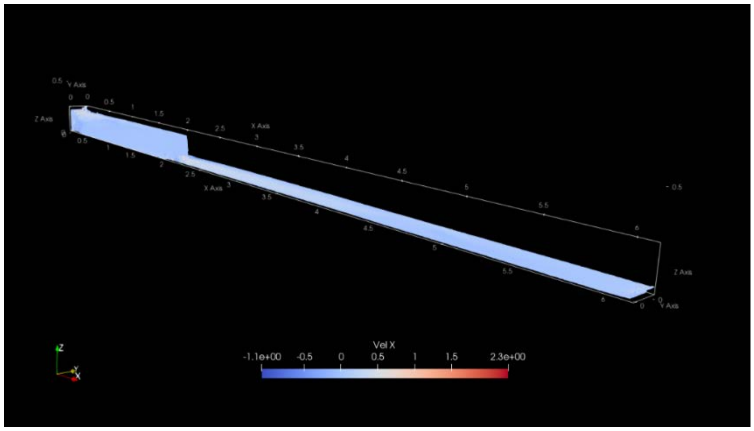


Figure C6. 3D Case. Steady State, colour plot of the x-velocity component. Only fluid particles are shown.

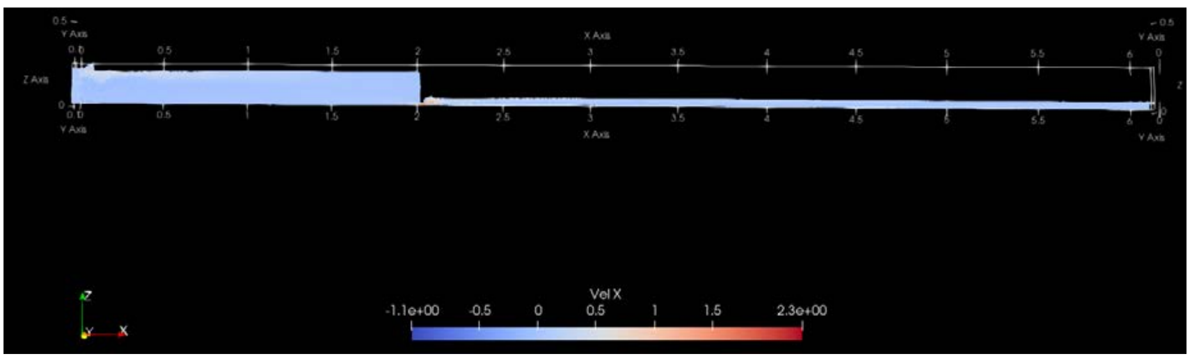


Figure C7. 3D Case. Steady state. Flow visualization at vertical mid-plane. Y=0.075 m.

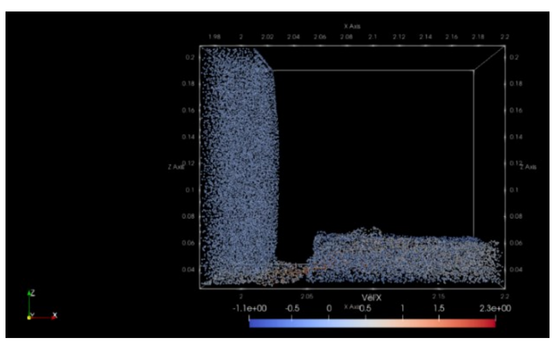


Figure C8. 3D Case. Steady State flow. Detail of flow under the sluice gate, side view. 1.97 ≤ x ≤ 2.2 m.

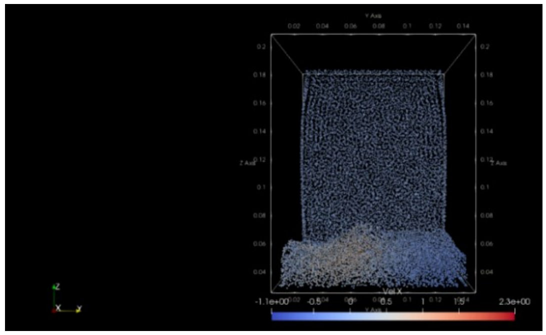


Figure C9. 3D Case. Steady state. Flow in the vicinity of the sluice gate, front view from plane x=2.12 m.

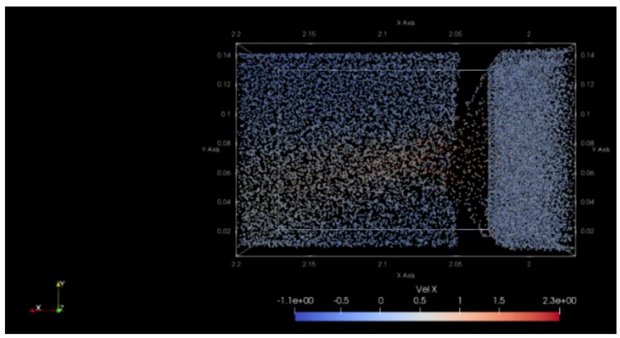


Figure C10. Case. Steady state flow top view for 1.97< x < 2.2 m

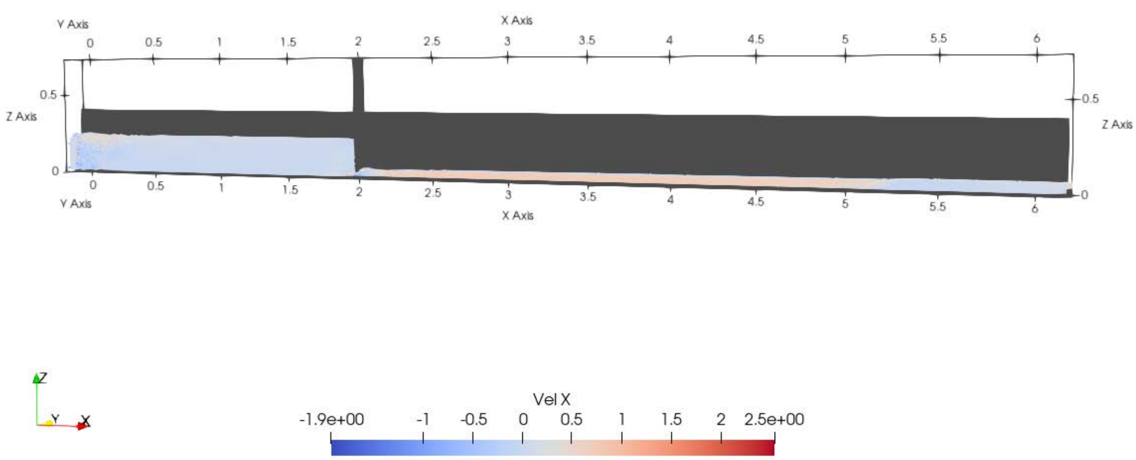


Figure C11. 3D Case. Steady State. Overall flow field at Y=0.07 m.

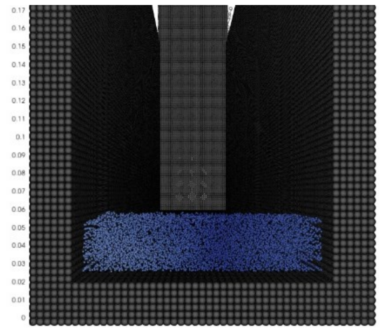


Figure C12. 3D Case. Steady state. Fluid particle distribution at x=3.2 m looking upstream. The sluice gate is visible at x=2.02 m.

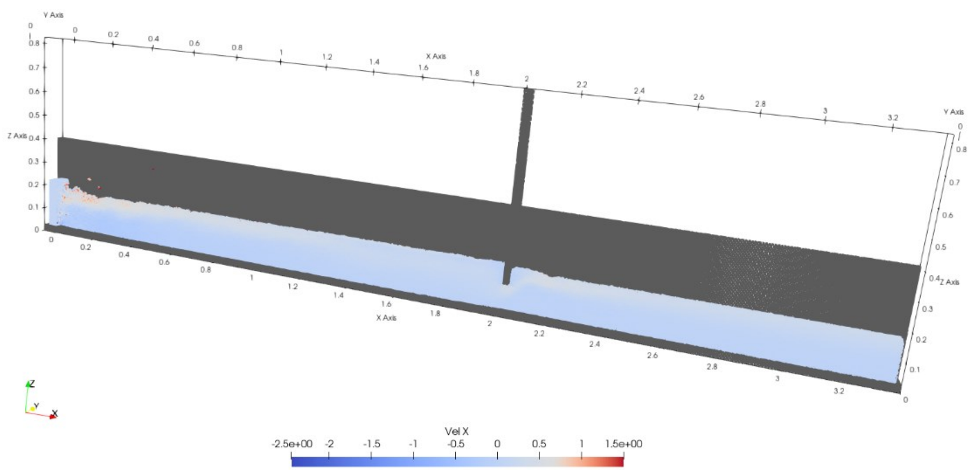


Figure C13. 3D Case. Steady State flow visualization at vertical mid-plane. Y=0.075 m, 0 ≤ x ≤ 3.5 m.

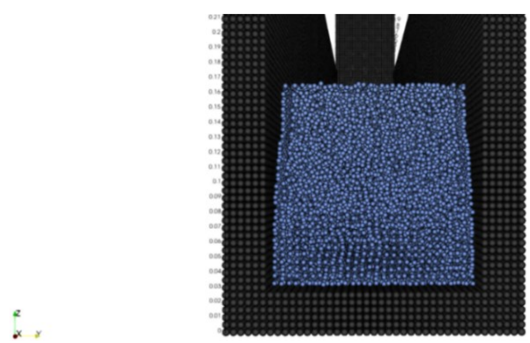


Figure C14. 3D Case. Steady state. Fluid particle distribution at x=3.2 m looking upstream. The sluice gate is visible at x=2.02 m.

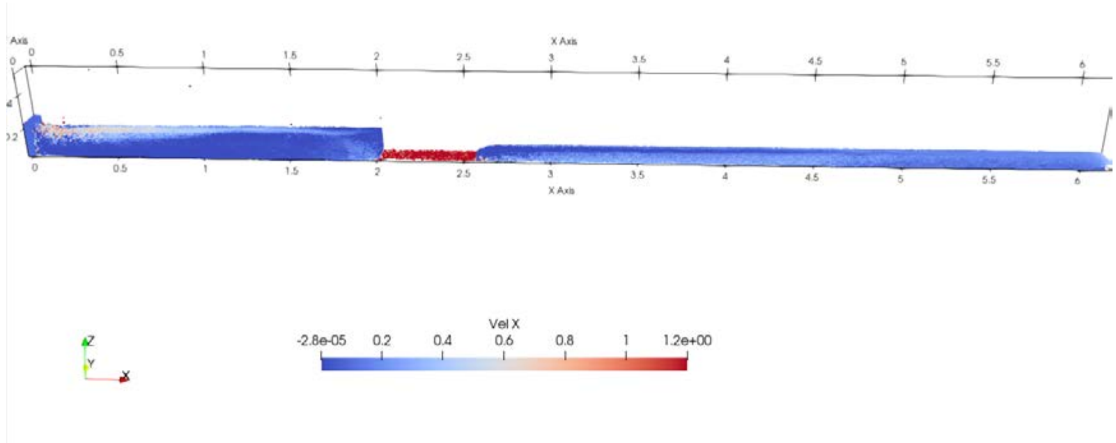


Figure C15. Case. Overall flow field.

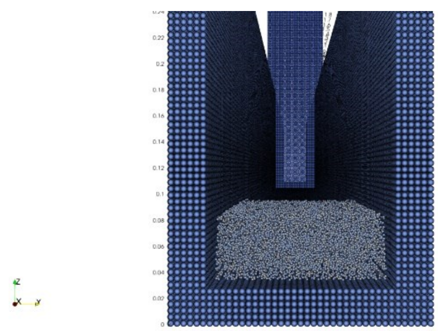


Figure C16. 3D Case. Steady state. Fluid particle distribution at x=4 m looking upstream. The sluice gate is visible at x=2.02 m.

D

Algorithm D1. Prepare variables for interaction functions.

```
void JSphCpu::PreInteractionVars_Forces(unsigned np,unsigned npb){
  //-Initialise arrays.
  const unsigned npf=np-npb;
  memset(Arc,0,sizeof(float)*np);                               //Arc[]=0
  if(Deltac)memset(Deltac,0,sizeof(float)*np);                  //Deltac[]=0
  memset(Acec,0,sizeof(tfloat3)*np);                            //Acec[]=(0,0,0)
  if(SpsGradvelc)memset(SpsGradvelc+npb,0,sizeof(tsymatrix3f)*npf);
//SpsGradvelc[]=(0,0,0,0,0,0,0).
 if(Dgradvelc)memset(Dgradvelc+npb,0,sizeof(tsymatrix3f)*npf);
//Dgradvelc[]=(0,0,0,0,0,0).
```

Algorithm D2. Free memory assigned to Arrays.

```
void JSphCpu::PosInteraction_Forces(){
  //-Free memory assigned in PreInteraction_Forces().
  ArraysCpu->Free(Arc);         Arc=NULL;
  ArraysCpu->Free(Acec);        Acec=NULL;
  ArraysCpu->Free(Deltac);      Deltac=NULL;
  ArraysCpu->Free(ShiftPosfsc);  ShiftPosfsc=NULL;
  ArraysCpu->Free(Pressc);      Pressc=NULL;
  ArraysCpu->Free(SpsGradvelc);  SpsGradvelc=NULL;
  ArraysCpu->Free(Dgradvelc);  Dgradvelc=NULL;
}
```

Algorithm D3. Perform interaction between ghost nodes and fluid particles. The same is for interaction between fluid and fluid particles.

```
//-Interaction of boundary with type Fluid/Float.
   bool rsym=false; //<vs_syymmetry>
   for(unsigned p2=pif.x;p2<pif.y;p2++){
    const float drx=float(posp1.x-pos[p2].x);
    float dry=float(posp1.y-pos[p2].y);
    if(rsym)    dry=float(posp1.y+pos[p2].y); //<vs_syymmetry>
    const float drz=float(posp1.z-pos[p2].z);
    const float rr2=drx*drx+dry*dry+drz*drz;
    if(rr2<=KernelSize2 && rr2>=ALMOSTZERO){
      //-Computes kernel.
          const float fac=fsph::GetKernel_Fac<tker>(CSP,rr2);
          const float frx=fac*drx,fry=fac*dry,frz=fac*drz; //-Gradients.
```

Algorithm D4. Wendland kernel calculation.

```
//-Wendland kernel.
     float fac;
     const float wab=fsph::GetKernel_WabFac<tker>(CSP,rr2,fac);
     const float frx=fac*drx,fry=fac*dry,frz=fac*drz; //-Gradients.
```

Algorithm D4. Modification of the Wendland kernel calculation with a new 2h smoothing length. This serves for the test filtering of the Dynamic Smagorinsky

```
// Adjust rr2 to use 2h as the smoothing length
const float rr2_new = rr2 / 4.0f;  // (2h)^2 = 4h^2
const float wab = fsph::GetKernel_WabFac<tker>(CSP, rr2_new, fac);
```

Algorithm D5. Calculation of the velocity gradients. Where dfrx, dfry, dfrz are the modified's kernel gradients.

```
//-Velocity gradients.
      if(!ftp1){//-When p1 is a fluid particle.
        const float volp2=-massp2/velrhop2.w;
float dv=dvx*volp2; gradvelp1.xx+=dv*frx; gradvelp1.xy+=dv*fry; gradvelp1.xz+=dv*frz;
    dv=dvy*volp2; gradvelp1.xy+=dv*frx; gradvelp1.yy+=dv*fry; gradvelp1.yz+=dv*frz;
    dv=dvz*volp2; gradvelp1.xz+=dv*frx; gradvelp1.yz+=dv*fry; gradvelp1.zz+=dv*frz;
tsymatrix3f dgradvelp1;
float dv_d=dvx*volp2; dgradvelp1.xx+=dv*dfrx; dgradvelp1.xy+=dv*dfry;
dgradvelp1.xz+=dv*dfrz;
dv_d=dvy*volp2; dgradvelp1.xy+=dv*dfrx; dgradvelp1.yy+=dv*dfry;
dgradvelp1.yz+=dv*dfrz;
dv_d=dvz*volp2; dgradvelp1.xz+=dv*dfrx; dgradvelp1.yz+=dv*dfry;
dgradvelp1.zz+=dv*dfrz;
```

Algorithm D6. Computation of the tau for SPS model.

```
/===========================================================================
/// Computes sub-particle stress tensor (Tau) for SPS turbulence model.
//===========================================================================
void JSphCpu::ComputeSpsTau(unsigned n,unsigned pini,const tfloat4 *velrhop,const tsymatrix3f *gradvel,const tsymatrix3f *dgradvel,tsymatrix3f *tau)const{
  const int pfin=int(pini+n);
  #ifdef OMP_USE
    #pragma omp parallel for schedule (static)
  #endif
  for(int p=int(pini);p<pfin;p++){
    const tsymatrix3f gradvel=SpsGradvelc[p];
    const tsymatrix3f dgradvel=Dgradvelc[p];
          const float pow1=gradvel.xx*gradvel.xx + gradvel.yy*gradvel.yy + gradvel.zz*gradvel.zz;  //Αθροισμα S_kk
      const float prr=pow1+pow1 + gradvel.xy*gradvel.xy + gradvel.xz*gradvel.xz + gradvel.yz*gradvel.yz; // Τα υπολοιπα στοιχεία του τανιστή (συμμετρικα) 2 S_ij*S_ij

const float powd1=dgradvel.xx*dgradvel.xx + dgradvel.yy*dgradvel.yy + dgradvel.zz*dgradvel.zz;  //Αθροισμα S_kk_testfilter
    const float prrd=powd1+powd1 + dgradvel.xy*dgradvel.xy + dgradvel.xz*dgradvel.xz + dgradvel.yz*dgradvel.yz; //test
   //=================================================================
/// Dynamic Smagorinsky
//=================================================================
   const float visc_sps=float(C_d)* sqrt(prr) ; //C_d is the new dynamic visco_sps
   const float div_u=gradvel.xx+gradvel.yy+gradvel.zz;
    const float ddiv_u=dgradvel.xx+dgradvel.yy+dgradvel.zz;
```

```
    const float sps_k=(2.0f/3.0f)*visc_sps*div_u;

    const float sps_blin=SpsBlin*prr;

    const float sumsps=-(sps_k+sps_blin);

    const float twovisc_sps=(visc_sps+visc_sps);

    const float one_rho2=1.0f/velrhop[p].w;

    tau[p].xx=one_rho2*(twovisc_sps*gradvel.xx +sumsps);

    tau[p].xy=one_rho2*(visc_sps   *gradvel.xy);

    tau[p].xz=one_rho2*(visc_sps   *gradvel.xz);

    tau[p].yy=one_rho2*(twovisc_sps*gradvel.yy +sumsps);

    tau[p].yz=one_rho2*(visc_sps   *gradvel.yz);

    tau[p].zz=one_rho2*(twovisc_sps*gradvel.zz +sumsps);

  }

}
```

E

Algorithm E1. In-house kernel library in Python. Kernels: Gaussian, Cubic Spline and Wendland.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.cm as cm
h = 1.0  # Smoothing length

# Gaussian Kernel
def W_gaussian(r):
    return 1/(np.pi * h**2) * np.exp(-(r/h)**2)

def dW_gaussian(r):
    return -2 * r/(np.pi * h**2) * np.exp(-(r/h)**2)

# Cubic Spline Kernel
def W_cubic(r):
    q = np.abs(r)/h
    if q >= 0 and q <= 1:
        return (10/(7*np.pi*h**2)) * (1 - 1.5*q**2 + 0.75*q**3)
    elif q <= 2:
        return (10/(7*np.pi*h**2)) * (0.25 * (2 - q)**3)
    else:
        return 0.0


def dW_cubic(r):
    q = np.abs(r)/h
    if q>=0 and q <= 1:
        val = -3*q + 2.25*q**2
    elif q <= 2:
        val = -0.75 * (2 - q)**2
    else:
        val = 0.0
    return (10/(7*np.pi*h**2)) * val * np.sign(r)

# Wendland C2 Kernel
def W_wendland(r):
    q = np.abs(r)/h
    if q>=0 and q <= 2:
        return (7/(4*np.pi*h**2)) * ((1 - 0.5*q)**4 * (2*q + 1))
    else:
        return 0.0


def dW_wendland(r):
```

```
    q = np.abs(r)/h
    if q>=0 and q <= 2:
        return (7/(4*np.pi*h**2)) * -5 * q * (1 - 0.5*q)**3 * np.sign(r)
    else:
        return 0.0

# Plotting 2D and 3D

def plot_kernel_and_derivative(kernel_func, derivative_func, title):
    r = np.linspace(-2, 2, 400)
    W = [kernel_func(ri) for ri in r]
    dW = [derivative_func(ri) for ri in r]
    plt.figure(figsize=(8, 6))
    plt.plot(r, W, label='Kernel W(r)', color='blue')
    plt.plot(r, dW, label='Derivative dW/dr', color='red', linestyle='--')
    plt.title(f'{title} - Kernel and Derivative')
    plt.xlabel('r/h')
    plt.ylabel('Value')
    plt.axhline(0, color='black', lw=0.5, ls='--')
    plt.grid(True)
    plt.legend()
    plt.show()

def plot_3d_kernel(kernel_func, title):
    x = np.linspace(-2, 2, 50)
    y = np.linspace(-2, 2, 50)
    X, Y = np.meshgrid(x, y)
    R = np.sqrt(X**2 + Y**2)
    Z = np.vectorize(kernel_func)(R)
    fig = plt.figure(figsize=(8, 6))
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, Z, cmap=cm.viridis)
    ax.set_title(title)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('W(r)')
    plt.show()

# Plotting all kernels
plot_kernel_and_derivative(W_gaussian, dW_gaussian, 'Gaussian Kernel')
plot_3d_kernel(W_gaussian, '3D Gaussian Kernel')

plot_kernel_and_derivative(W_cubic, dW_cubic, 'Cubic Spline Kernel')
plot_3d_kernel(W_cubic, '3D Cubic Spline Kernel')

plot_kernel_and_derivative(W_wendland, dW_wendland, 'Wendland C2 Kernel')
plot_3d_kernel(W_wendland, '3D Wendland C2 Kernel')
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΕΥΣΤΑΘΙΟΣ ΧΑΤΖΟΓΛΟΥ

ΒΟΛΟΣ 2025