2021-03-15

High-fidelity numerical simulation of solitary wave propagation

Huang, Y

http://hdl.handle.net/10026.1/17646

Highlights

High-fidelity numerical simulation of solitary wave propagation

Yichen Huang,Jing Li,Bin Xie,Zhiliang Lin,Alistair G. L. Borthwick

- High-fidelity simulations undertaken of solitary wave propagation on plane and sloping beaches

- A sophisticated Navier–Stokes solver is developed on polyhedral unstructured meshes

- Numerical dissipation and dispersion controlled by high-order reconstruction

- Model predictions significantly reduce inaccuracies exhibited as spurious energy loss, wave decay, phase shift, and overestimated velocity profile

High-fidelity numerical simulation of solitary wave propagation

Yichen Huang[a], Jing Li[a], Bin Xie[a], Zhiliang Lin[a,b,*] and Alistair G. L. Borthwick[c]

[a]*State Key Laboratory of Ocean Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

[b]*Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration, Shanghai 200240, China*

[c]*School of Engineering, The University of Edinburgh, Edinburgh EH9 3JL, United Kingdom*

ARTICLE INFO

ABSTRACT

As a classic problem in oceanographic and coastal engineering hydrodynamics, solitary wave propagation is used to benchmark a novel high-fidelity numerical model based on the Navier–Stokes equations. The model combines the Finite Volume method based on Merged Stencil with 3rd-order reconstruction (FVMS3) and Tangent of Hyperbola for INterface Capturing method with Quadratic surface representation and Gaussian Quadrature (THINC/QQ) schemes. Accurate predictions are made of inviscid and viscous solitary waves propagating on plane and sloping beaches. Model performance is assessed by comparing the predictions with analytical solutions, alternative numerical results obtained using interFoam solver in OpenFOAM and commercial flow software ANSYS Fluent, as well as experimental data. It is demonstrated that the present model significantly reduces undesirable numerical effects, including energy loss, wave decay, phase shift, and overestimation of the velocity profile of propagating waves even after long duration computations, by suppressing numerical dissipation and dispersion.

1. Introduction

Free surface flow simulations based on computational fluid dynamics (solving the fundamental continuity and Navier-Stokes momentum equations) are nowadays routinely used in oceanography and coastal engineering. Example applications include the propagation and breaking of solitary waves over submerged obstacles (Lubin, 2004), the modeling of fluid-structure interactions in free surface flows (Formaggia, Miglio, Mola and Parolini, 2008), and simulations of waves generated by rigid and deforming or subaerial landslides (Abadie, Morichon, Grilli and Glockner, 2010; Montagna, Bellotti and Di Risio, 2011), all of which require consideration of viscous effects and deformation of the free surface. In such cases, an immiscible interface separates two fluids, air and water, with very different physical properties. The interface poses substantial challenges for numerical methods required to resolve the free surface and associated sudden jump in fluid properties, such as density. However, the majority of industry-standard solvers for viscous wave problems have been developed using second-order accurate schemes which are afflicted by excessive numerical dissipation and dispersion, causing severe wave decay and phase shift errors. For example, simulations of regular waves using STAR-CCM+ software and interFoam solver have been demonstrated to be very sensitive to temporal and spatial resolution (Perić and Abdel-Maksoud, 2015; Zhang, Zhao, Xie and Nie, 2019). Many Navier–Stokes solvers, including Gerris, interFoam, Thétis and Truchas, have made serious overestimates of wave velocity profiles (Wroniszewski, Verschaeve and Pedersen, 2014). Despite recent progress, high-fidelity numerical solvers that precisely resolve free surface wave propagation in complicated coastal domains are far from well-established.

Among computational fluid dynamics (CFD) solvers, models based on a combination of the Finite Volume Method (FVM) and Volume of Fluid (VOF) method have become particularly popular because of their conservative property and ease of implementation on adaptive, unstructured meshes (Tryggvason, Scardovelli and Zaleski, 2011). In the VOF method, the moving interface is either explicitly created as a line (Youngs, 1982; Parker and Youngs, 1992; Aulisa, Manservisi, Scardovelli and Zaleski, 2003) or approximated by addition of artificial compression (Harten, 1978; Ubbink and Issa, 1999; Xiao and Ikebata, 2003). The latter approach, which does not involve complicated geometric calculations, is more appropriate for unstructured meshes, but at reduced numerical accuracy.

The Tangent of Hyperbola for INterface Capturing method with Quadratic surface representation and Gaussian Quadrature (THINC/QQ) represents curved surfaces by a complete quadratic function (Xie and Xiao, 2017). THINC/QQ

is particularly attractive in that it can accurately handle topologically changing interfaces on unstructured meshes without loss of both efficiency and simplicity. In computational fluid dynamics, higher-order methods are preferable because their truncation errors decrease rapidly on refined meshes, thus alleviating undesirable effects such as wave decay and phase shifts. Recently, the Finite Volume method based on Merged Stencil with 3rd-order reconstruction (FVMS3) was proposed for spatial discretization of the fundamental fluid flow equations, achieving third-order accuracy on polyhedral unstructured meshes (Xie, Deng and Liao, 2019).

To obtain high-fidelity wave propagation simulations, the present paper utilizes a sophisticated numerical model constructed on polyhedral unstructured meshes, which combines FVMS3 and THINC/QQ schemes. The resulting MultiphAse solver based on high-fidelity Reconstruction SchemeS is named MARS2, which is implemented by using an open-source library of OpenFOAM. (The reviewer assumes this solver is developed based on OpenFoam by reading the governing equations, especially the momentum equation.) As a benchmark test, the propagation of solitary waves in water of otherwise uniform depth is used to test the performance of MARS2, given the substantial body of theoretical and experimental research studies on this classic phenomenon (Liu and Gollub, 1994; Bona, Pritchard and Scott, 1980; Ablowitz, Kruskal and Ladik, 1979; Munk, 1949; Maxworthy, 1976; Lo and Shao, 2002; Wroniszewski et al., 2014). As demonstrated later, the proposed model possesses considerable advantages in controlling numerical dissipation and dispersion, and in suppressing wave decay and spurious phase shifts. MARS2 is then used to study solitary wave runup, with and without viscosity present; the model is found to give accurate simulations of wave runup and rundown in close agreement with predictions by Wroniszewski et al. (2014) and experimental measurements by Lin, Chang and Liu (1999).

The paper is organized as follows. Section 2 outlines the governing equations, and describes the numerical solver. Section 3 presents a summary of the test cases. Section 4 interprets the numerical simulations, with comprehensive comparisons against alternative predictions by MARS2, interFoam and Fluent solvers, validated by analytical solutions and experimental data. Section 5 summarizes the key findings.

2. Multiphase solver based on high-fidelity reconstruction schemes

2.1. Navier–Stokes equations with VOF function

Consider the two-dimension free water surface flow problem described by the incompressible species advection, continuity, and Navier–Stokes momentum equations:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \phi \nabla \cdot \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + (\nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \mathbf{u} \cdot \nabla \mu) - \mathbf{g} \cdot \mathbf{x}\nabla\rho, \tag{3}$$

where $\phi(\mathbf{x}, t)$ is an indicator function used to track the moving interface, $\mathbf{u} = (u, v)$ is the velocity field, ρ the fluid density, p the modified pressure obtained by subtracting the hydrostatic pressure from the total pressure, $\mathbf{g}$ the acceleration due to gravity, μ the fluid dynamic viscosity, and $\otimes$ the Kronecker product.

Surface tension is neglected because it has hardly any impact on solitary wave motion due to the extremely large wavelength involved. In our work, a volume fraction or VOF function $\phi(\mathbf{x}, t)$ is used to distinguish two immiscible fluids, i.e. air and water, by

$$\phi(\mathbf{x}, t) = \begin{cases} 1 & water \\ 0 & air \\ 0 < \phi < 1 & interface. \end{cases} \tag{4}$$

In the one-fluid model, the dynamic effects of intrinsic fluid properties, such as density and viscosity, are updated according to the volume fraction as,

$$\rho = \rho_1 \phi + \rho_2 (1 - \phi), \tag{5}$$

$$\mu = \mu_1 \phi + \mu_2 (1 - \phi), \tag{6}$$

where ρ_1, ρ_2 are the densities and μ_1, μ_2 are the dynamic viscosity coefficients of water and air respectively.

2.2. FVMS3 scheme

The Finite Volume method based on Merged Stencil with 3rd-order reconstruction (FVMS3) used for spatial discretization of the governing Navier–Stokes equations, is third-order accurate on polyhedral unstructured meshes (Xie et al., 2019). To illustrate this, a brief introduction to FVMS3 for solving 2D problems follows.

First, the computational domain is divided into non-overlapping control volumes Ω_i ($i = 1, ..., I$). Fig. 1(a) shows a two-dimensional polygon with an arbitrary number of vertices $\theta_{ik}(k = 1, ..., K)$ and boundary segments $\Gamma_{ij}(j = 1, ..., J)$, where K and J denote the total number of cell vertices and surfaces. The outward normal unit vector on the boundary surface is denoted $\mathbf{n}_{ij}$. The coordinates of vertex θ_{ik} and mass center θ_i^c are denoted (x_{ik}, y_{ik}) and (x_i^c, y_i^c), respectively. The boundary area magnitude is $|\Gamma_{ij}|$ and mass volume is $|\Omega_i|$.

Within any arbitrary mesh element Ω_i, all the computational variables $\bar{\varphi}_i(t)$ are discretized by the volume integrated average (VIA) as

$$\bar{\varphi}_i(t) \equiv \frac{1}{|\Omega_i|} \int_{\Omega_i} \varphi(x, y, t) \mathrm{d}\Omega, \tag{7}$$

where $\varphi(x, y, t)$ represents the physical fields, comprising velocity $\mathbf{u}$, pressure p, and VOF function ϕ.

Given the volume integrated average of each cell, the discrete variables φ can be reconstructed in a piecewise fashion using the quadratic polynomial,

$$\Phi_i(\varphi; x, y, t) = \sum_{p=0}^{2} \sum_{q=0}^{p+q \leq 2} \left(x - x_i^c\right)^p \left(y - y_i^c\right)^q c_{pq}, \tag{8}$$

on an admissible stencil (see Fig. 1(b)) produced by merging all neighboring cells $\Omega_{ikl}(l = 1, ..., L)$ around the joint vertices θ_{ik} of the target cell Ω_i. The unknown coefficients c_{pq} in Eq. (8) are determined by the least-squares method.
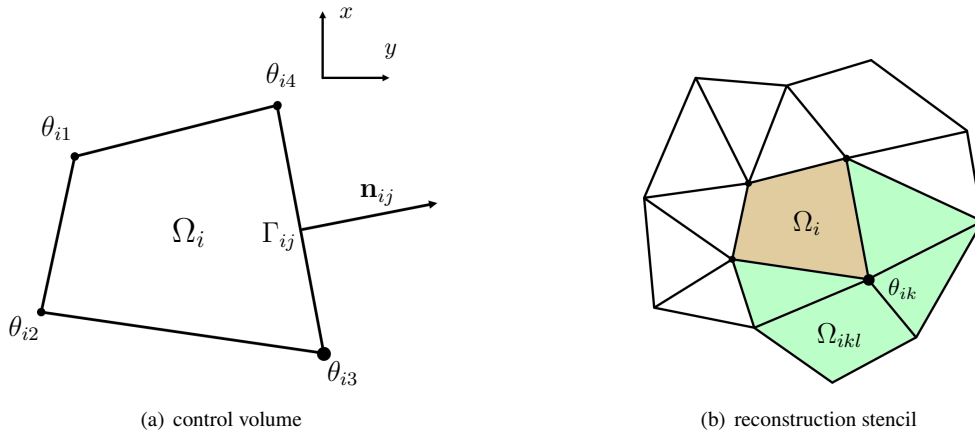

Figure 1: Definition sketch showing control volume (a) and reconstruction stencil (b) used for 2D unstructured mesh composed of polyhedral elements.

2.3. THINC/QQ scheme

For free surface flow problems, the THINC/QQ scheme is used to represent curved surfaces by a complete quadratic function. To reconstruct the VOF function, the hyperbolic tangent function $H_i(x, y)$ in global coordinates is given as:

$$H_i(x, y) = \frac{1}{2} \left(1 + \tanh \left(\beta \left(P_i(x, y) + d_i\right)\right)\right), \tag{9}$$

where β is a steepness parameter specified to control the thickness of the transition area with a properly bounded volume fraction.

The interface equation $P_i(x, y) + d_i = 0$ is approximated by a fully quadratic polynomial whose coefficients are retrieved from Eq. (8). The sole unknown is d_i in Eq. (9), indicating that the interface location is uniquely determined from VIA in the target cell $\bar{\phi}_i$:

$$\frac{1}{|\Omega_i|} \int_{\Omega_i} H_i(x, y) \mathrm{d}x\mathrm{d}y = \bar{\phi}_i, \tag{10}$$

where the volume integration of the hyperbolic tangent function $H_i(x, y)$ is approximated using a Gaussian quadrature scheme. The VOF function is then updated using the following finite volume formulation:

$$\frac{\partial \bar{\phi}_i(t)}{\partial t} + \frac{1}{|\Omega_i|} \sum_{j=1}^{J} \left(v_{n_{ij}} \bar{\phi}_{ij} |\Gamma_{ij}| \right) = \frac{\bar{\phi}_i(t)}{|\Omega_i|} \sum_{j=1}^{J} (v_{n_{ij}} |\Gamma_{ij}|), \tag{11}$$

where $v_{n_{ij}}$ is the transport velocity through the cell boundary Γ_{ij} satisfying the divergence-free condition. The surface average $\bar{\phi}_{ij}$ is obtained by integrating the reconstruction function $H_{\mathrm{up}}(x, y)$ in the upwinding cell at the surface Γ_{ij},

$$\bar{\phi}_{ij} = \frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} H_{\mathrm{up}}(x, y) \mathrm{d}\Gamma, \tag{12}$$

where the subscript up is an upwinding index of neighboring cells across the boundary surface Γ_{ij}.

2.4. Solution procedure of MARS2

By combining the FVMS3 and THINC/QQ schemes for solving the fundamental fluid flow equations along with a VOF function, a new framework called MARS2 (MultiphAse solver based on high-fidelity Reconstruction SchemeS) is proposed for the accurate resolution of interfacial multiphase flows on polyhedral unstructured meshes. In addition to its high-order reconstruction schemes, MARS2 uses a consistent formulation of the mass and momentum transport equations to cope with multiphase flows of large density ratios, which approximates the convective fluxes of volume fraction and momentum with identical discrete schemes. Owning to the rigorous mass conservation property of VOF scheme, the conservation of total mass of MARS2 is properly preserved throughout the numerical simulations. It also employs a balanced-force formulation which enforces identical discretization for gradient operators at cell boundary to achieve a correct balance between pressure and gravity forces. Moreover, a limiting strategy for the multi-dimensional hyperbolic conservation laws on unstructured grids called MPLu2 (Park and Kim, 2010) is used with the smoothness indicator function after computing convection fluxes(term), to prevent spurious velocity oscillations, especially in the vicinity of the air-water interface. See Xie, Jin, Du and Liao (2020) and Xie et al. (2019) for more details. (Also, third-order scheme is opt to cause spurious problem for computing convection fluxes, especially in the vicinity of the air-water interface).

MARS2 utilises the fraction-step approach (Chorin, 1968; Harlow and Welch, 1965; Kim and Moin, 1985) along with the third-order Total Variation Diminishing (TVD) Runge-Kutta scheme (Shu and Osher, 1988) to update the numerical solution from time level n $(t = t^n)$ to $n + 1$ $(t = t^n + \Delta t)$. The semi-discrete formulations at each sub-stage are summarized as follows:

1. Given the volume fraction ϕ^n and velocity field $\mathbf{u}^n$, we solve the VOF advection equation to obtain ϕ^{n+1} and advance the velocity to $\mathbf{u}^*$ by solving the momentum equation using an explicit scheme,

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot (\mathbf{u}\phi) + \phi \nabla \cdot \mathbf{u}, \tag{13}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} = -\nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) + (\nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \mathbf{u} \cdot \nabla \mu) - \mathbf{g} \cdot \mathbf{x} \nabla \rho, \tag{14}$$

where density ρ and dynamic viscosity μ are updated using (5) and (6).

2. To satisfy the divergence-free condition (2), the pressure Poisson equation is derived from the incompressibility condition,

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p^{n+1} \right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \tag{15}$$

3. The velocity field is corrected to $\mathbf{u}^{n+1}$ by projecting the pressure field,

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}(\nabla p^{n+1}). \tag{16}$$

The updated velocity $\mathbf{u}^{n+1}$ satisfies the divergence-free condition (2), and is then used as the transport velocity for the next time level iteration.

3. Numerical tests

Two different series of numerical tests were undertaken: solitary wave propagation; and solitary wave runup. This section describes the computational models set up for both series of tests.

3.1. Propagation of a solitary wave

The first benchmark test comprises solitary wave propagation over a horizontal bed in a flume. The mean water depth d is set to 1 m and the initial position of the wave crest from the wavemaker boundary is $x_0 = 15$ m. Two different solitary waves are considered, with amplitude to depth ratio H/d equal to 0.1 and 0.3 respectively. Table 1 lists the physical parameters, and resolutions of three meshes used for convergence tests related to each solitary wave of interest. An adaptive time step is utilised, with the maximum Courant number set to 0.5 for $H/d = 0.1$ and 0.2 for $H/d = 0.3$, if not otherwise specified. In both cases, the simulation duration was set to 12 s. Fig. 2 shows the evolution of the smaller solitary wave in space and time, on the finest mesh.

Table 1
Domain parameter settings and mesh resolutions for the solitary wave propagation simulations.

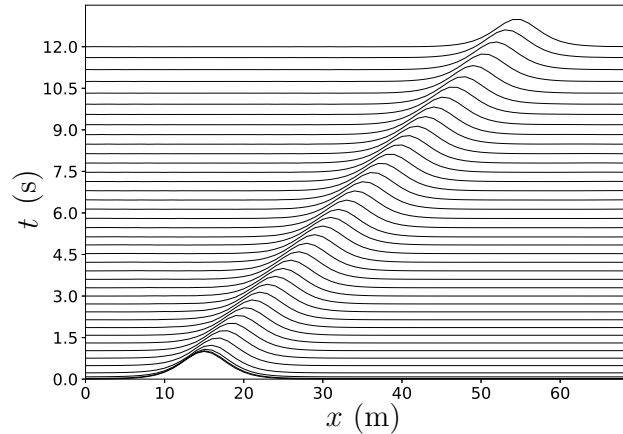| H/d	$L_x \times L_y$	Resolution 1	Resolution 2	Resolution 3
0.1	69 m × 1.5 m	1472 × 32	2944 × 64	5888 × 128
0.3	72.6 m × 2.2 m	1056 × 32	2112 × 64	4224 × 128

Figure 2: Space-time plot illustrating solitary wave propagation for $H/d = 0.1$.

To examine the performance of the MARS2 solver in terms of (mass and) energy conservation, wave-shape preservation, and velocity profile accuracy, we carry out a comparison with corresponding results from the open-source code interFoam, software Fluent with third-order MUSCL discretization scheme and the analytical solution given by potential flow theory. (According to the reviewer's experience, there are some open source flow solvers and commercial flow software packages within which there are options for third-order or higher-order discretization scheme for convection terms. For instance, Ansys-Fluent has third-order schemes available, and Gerris has high-order compact

schemes available.) Here, an (inviscid) Euler formulation, with slip boundary condition at the bed and end periodic boundary conditions, is used to facilitate comparison against the well-established analytical solution.

Simulations of a solitary wave of $H/d = 0.1$ were run on regular, stretched, and hybrid meshes in order to explore the possible influence of different meshing strategies on the results. Each mesh is two-dimensional in the $x - y$ plane, where x is the horizontal distance along the flume, and y is the vertical distance above the bed. Fig. 3(a) shows an example of a regular mesh. Fig. 3(b) displays a typical stretched mesh obtained by concentrating nodes at still water level; this refinement in the vicinity of the air-water interface is analogous to capturing the free surface. Meanwhile, the mesh density is coarser at the upper and lower boundaries in order to maintain the same total number of cells as in the regular mesh cases. Fig. 3(c) shows the hybrid mesh formed by fitting triangular cells to the stretched mesh, again refined in the region of wave propagation. The hybrid mesh provides an essential test by which to examine the applicability of solvers for complex geometrical configurations, noting that the computation of fluid motion at mesh intersections is particularly challenging.
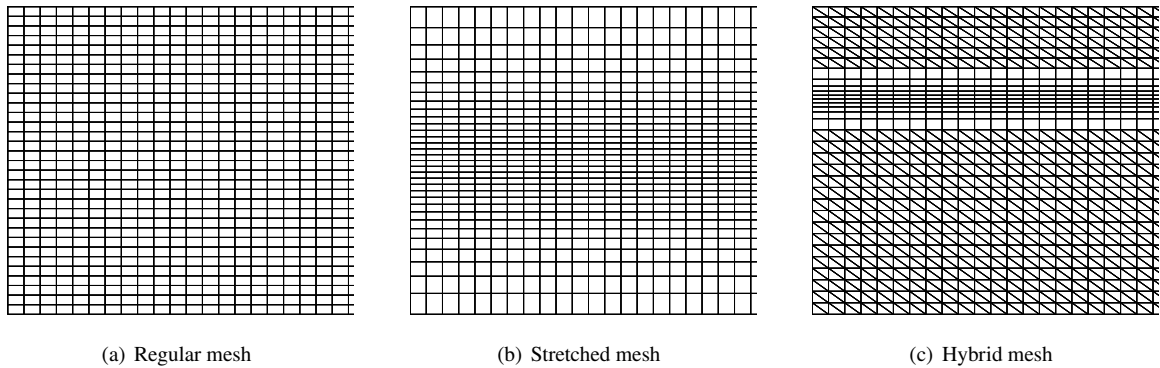

(a) Regular mesh   (b) Stretched mesh   (c) Hybrid mesh

Figure 3: Three types of meshes used in the solitary wave propagation simulations for $H/d = 0.1$. Mesh (b) is graded with doubly increasing resolution.

3.2. Runup of a solitary wave

In the second series of tests, MARS2 and interFoam are used to simulate the runup at a plane beach of two inviscid non-breaking solitary waves with relative amplitudes H/d of 0.1 and 0.3 respectively. The initial position of each solitary wave is 15 m in front of the toe of the bed slope. Two beach slopes are considered: 10° and 30°. For comparison purposes, the geometric and computational conditions are identical to those of Wroniszewski et al. (2014). The time-step is adaptively adjusted so that the maximum Courant number is 0.2. Convergence tests are again conducted on meshes of different resolution. Fig. 4 presents snapshots of the propagation and runup of a solitary wave with $H/d = 0.3$ on the 10° sloping beach at times corresponding to: the initial conditions; when the solitary wave is partway up the beach; and at almost maximum runup.

In order to compare the model simulations with results from laboratory experiments described by Lin et al. (1999), three-dimensional runup simulations are conducted by MARS2, as well as two-dimensional simulations by MARS2 and interFoam for exploring whether there exist 3D effects in wave simulations. The fluid kinematic viscosity is set to $\nu_{water} = 10^{-6}$ m^2/s and $\nu_{air} = 1.48 \times 10^{-5}$ m^2/s, and a no-slip boundary condition imposed at the bed. Fig. 5 shows the two-dimensional numerical wave tank which is 8.99 m long and 3 m high with a 30° sloping beach commencing 8.49 m from the origin (the width of water tank is set to 0.6 m in 3D case). To eliminate any influence of the inlet boundary, the initial position of the wave crest is set at $x_0 = 2.0$ m. The undisturbed water depth d is 0.16 m, and the initial solitary wave height H is 0.027 m such that $H/d = 0.16875$. For comparison against the experimental data, velocity profiles through the water depth are recorded at 3 locations: $x = 8.3972, 8.5556$ and 8.7146 m ($z = 0.3$ m in 3D simulation). A schematic diagram for two-dimensional mesh structure in both runup simulations is presented in Fig. 6. All meshes utilized in this paper are produced by Gmsh.
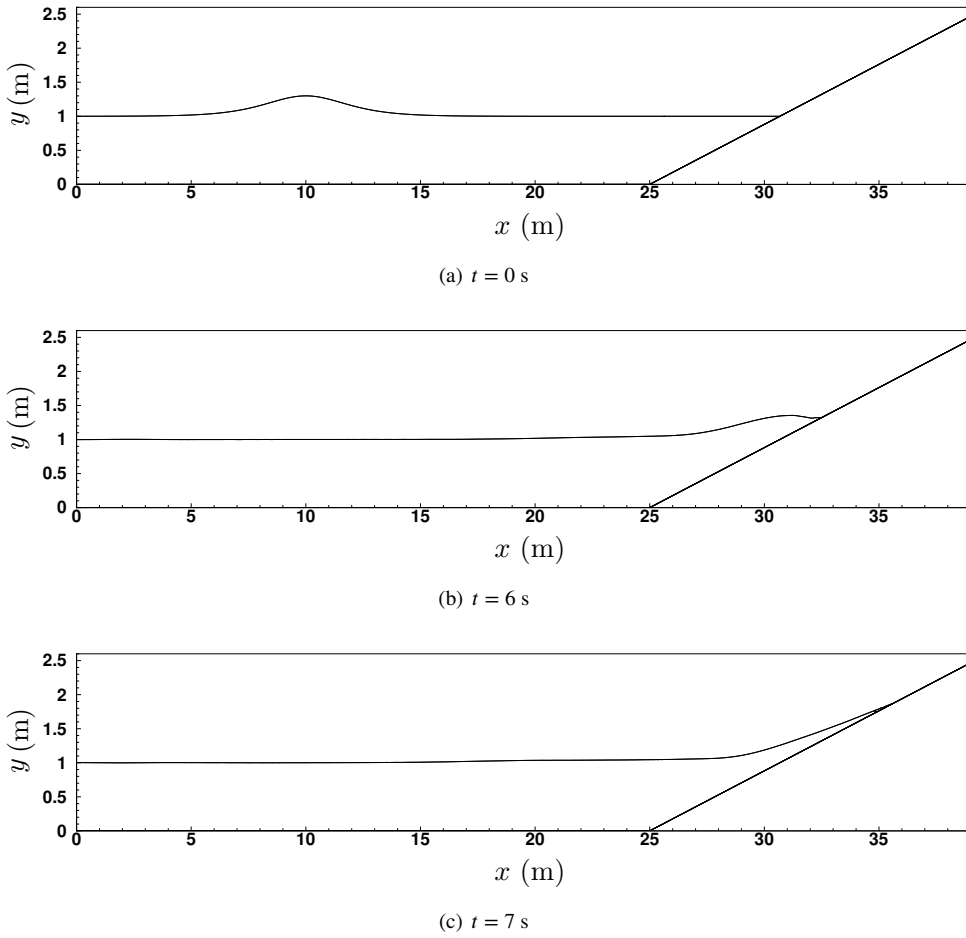
(a) $t = 0$ s

(b) $t = 6$ s

(c) $t = 7$ s

Figure 4: Free surface elevation profiles depicting the propagation and runup of a solitary wave of $H/d = 0.3$ on a $10°$ sloping beach at times $t = 0$, 6 and 7 s.
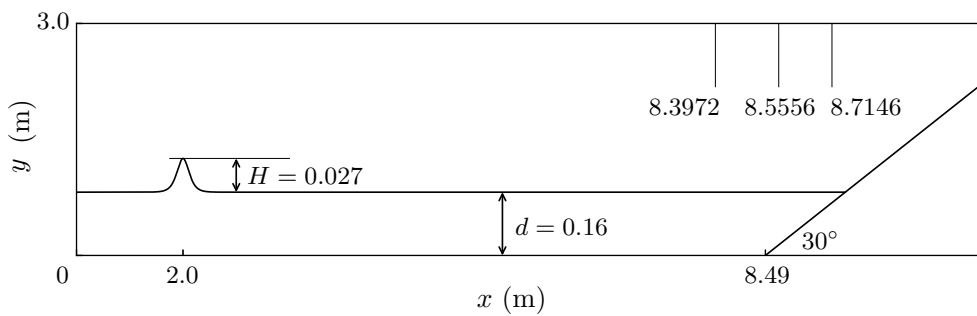

Figure 5: Definition sketch for runup and rundown of a solitary wave with $H/d = 0.16875$ on a $30°$ sloping beach.

4. Results and discussion

4.1. Propagation of a solitary wave on a regular mesh

In this section, a solitary wave propagating over a horizontal bed is first simulated on regular meshes of different resolution, and the predictions examined in terms of conservation of (mass and) total energy, deformation of waveform, and accuracy of velocity profile.

Figure 6: Schematic diagram for mesh structure in wave runup simulations.

4.1.1. Conservation of (mass and) total energy

Table 2 lists the analytical, initial values of kinetic, potential, and total energy, calculated using Laitone's second-order theory (Laitone, 1960) according to Eq. (25). Total energy remains unchanged over time regardless of fluid viscosity and the air phase effects, and so is designated as the exact reference solution.

Table 2
Reference values of mechanical energy of a solitary wave, calculated using Laitone's 2nd theory (Laitone, 1960).

H/d	Kinetic energy (J)	Potential energy (J)	Total energy (J)
0.1	257.4672	247.2084	504.6756
0.3	1587.4579	1396.4102	2983.8681

It has previously been established (Wroniszewski et al., 2014) that conventional second-order finite volume solvers suffer from excessive numerical dissipation, leading to noticeable decrease in predicted wave energy even for an inviscid liquid. Besides, the third-order MUSCL scheme is found to undergo more dissipation and non-physical oscillation around the discontinuities (Leng, Li, Fu and Ma, 2012). We therefore examine the time history of predicted total energy normalized against the corresponding analytical value (such that the accurate nomalized reference energy is unity).

The initial value of total energy integrated over the discretized domain approaches the analytical solution with mesh refinement, confirming reliable initialization during preprocessing. Fig. 7 shows that the total energy predictions by the present, interFoam and Fluent solvers converge to the reference value as the mesh resolution increases. It also indicates that the total wave energy declines more rapidly with time on the coarser meshes particularly for interFoam and Fluent solvers.
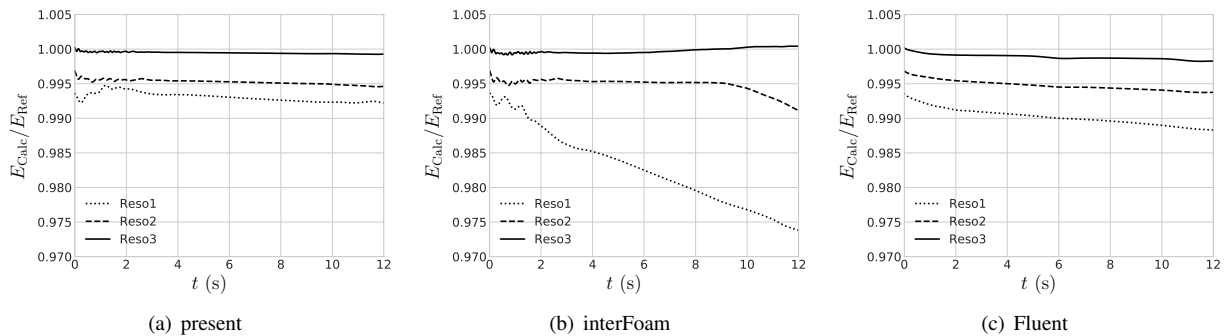

Figure 7: Predicted, normalized, total energy time histories for a solitary wave with $H/d = 0.1$ on regular meshes of resolution: 1472×32 (Reso1); 2944×64 (Reso2); and 5888×128 (Reso3).

It is obvious that the present predictions lie closer to the reference solution than those of interFoam and Fluent. Specifically, interFoam suffers from severe energy loss on the coarser meshes. At coarsest resolution, the normalized total energy predicted by interFoam declines rapidly, and exceeds 2.6% by $t = 12$ s. On the other hand, the maximum value of energy loss is merely 0.8% using the present solver, and 1.2% using Fluent, which are less than half of their

counterpart using interFoam. At double the mesh resolution, the normalized total energy predicted by interFoam remains almost steady until about $t = 9$ s, when it abruptly starts falling. In the finest mesh, the interFoam results present the opposite behaviour with the normalized total energy beginning to rise slightly after about $t = 5$ s. In contrast, the energy calculated by Fluent exhibits a monotonous decrease at all three resolutions, with the maximum energy loss surpass 2.5 times of MARS2 in the finest mesh (MARS2: 0.07%; Fluent: 0.17%). It can reasonably be supposed that, owing to Fluent's intrinsic numerical dissipation, the total energy of solitary wave predicted by Fluent would continue to largely reduce even if a more refined mesh is applied. Fig. 8 presents a close-up comparison between the results obtained on the finest mesh. Both the present and interFoam schemes exhibit rapid small-scale oscillations near the initial time, perhaps reflecting the numerical schemes adjustment to the analytical start conditions; while the Fluent solver experiences a rapid drop during this stage. The normalized total energy predicted by the present simulation then stabilises and declines slightly whereas the interFoam results experience a physically spurious increase in total energy (which is erroneous, noting that "since the solvers simulate also the flow in the air phase, transfer of momentum and energy to the air phase might add to the numerical dissipation" (Wroniszewski et al., 2014)). We should therefore be especially careful to bear this anomaly in mind when using interFoam to compute solitary wave propagation. Fluent continues to suffer from energy dissipation severely in spite of employing a third-order scheme, with relative error even larger than interFoam.

The MARS2 predictions outperform those of the interFoam and Fluent solvers. Fig. 7 and Fig. 8 show that the present total energy predictions fluctuate within reasonable limits allowing for slight attenuation. Excessive numerical dissipation on coarse meshes and growth in total energy on the finest mesh limit the reliability of interFoam solver for high-fidelity simulations of solitary wave propagation. For Fluent's 3rd-order MUSCL scheme, the tendency of energy dissipation is inevitable regardless of the mesh refinement.
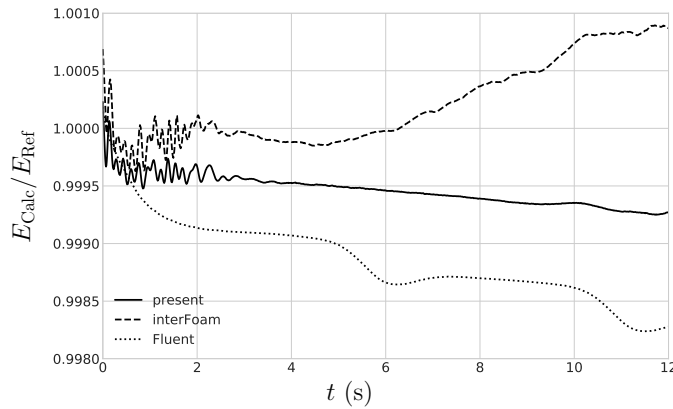

Figure 8: Predicted, normalized, total energy time histories for a solitary wave with $H/d = 0.1$ on a 5888 × 128 mesh (Resolution 3).

For the larger solitary wave where $H/d = 0.3$, Fig. 9 shows that the present simulations conserve total energy much better than interFoam and Fluent (whose predictions invariably experience energy decay with time). The error in total energy at $t = 12$ s predicted by interFoam is about 7.5% on the 1056 × 32 (Resolution 1) mesh. This error is about 5 times that using MARS2 and Fluent and 2.4 times that for $H/d = 0.1$ using interFoam. By contrast, the maximum decrease in normalized total energy at $t = 12$ s is approximately 1.7% using MARS2, a mere 1.3 times that for $H/d = 0.1$. Although the total energy calculated by Fluent is evidently more accurate than interFoam, it is inferior to MARS2 in every resolution. MARS2 is more accurate than interFoam and Fluent at conserving total energy of a propagating solitary wave, through its use of high-order reconstruction schemes that limit numerical dissipation.

4.1.2. Wave decay and phase shifts

Table 3 lists the predicted wave height and position of the wave crest relative to the wave maker of the solitary wave with $H/d = 0.1$ at $t = 10$ s.

Noting that the fluid viscosity is initially specified to be zero, the change in wave height with respect to the initial value acts as a proxy for numerical dissipation. Table 3 shows that the wave height prediction by MARS2 is almost
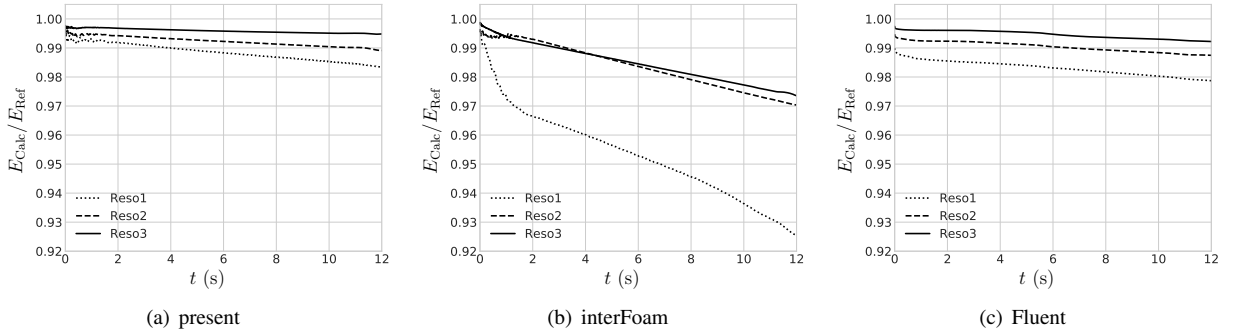
Figure 9: Predicted, normalized, total energy time histories for a solitary wave with $H/d = 0.3$ on regular meshes of resolution: 1472×32 (Reso1); 2944×64 (Reso2); and 5888×128 (Reso3).

Table 3
Wave height and crest location of solitary wave with $H/d = 0.1$ at $t = 10$ s.

Resolution	Wave height H (m)			Position of wave crest x (m)		
	1	2	3	1	2	3
Reference value	0.1			47.8400		
MARS2	0.0998	0.0997	0.1000	47.9531	47.9063	47.8711
interFoam	0.0973	0.0984	0.0990	47.7656	47.7422	47.7070
Fluent	0.0995	0.0994	0.0998	47.7188	47.6719	47.7305
	$(H_{ref} - H_{num})/H_{ref} \times 100\%$			$(x_{ref} - x_{num})/x_{ref} \times 100\%$		
MARS2	0.1983	0.2593	-0.0303	-0.2364	-0.1386	-0.0650
interFoam	2.7355	1.5608	0.9809	0.1555	0.2044	0.2780
Fluent	0.5288	0.5806	0.2484	0.2534	0.3514	0.2290

identical to the reference value on the finest mesh, with an error that is much smaller (0.03%) than that using inter-Foam (0.98%) and Fluent (0.25%). This confirms that MARS2 incurs almost no numerical dissipation (and hence no wave decay during solitary wave propagation over a flat, frictionless bed) compared to these two solvers. From Table 3, we also see that all solvers present a mismatch in wave phase between their numerical predictions and the exact solution. Solitary waves predicted by MARS2 travel a little faster than the analytical solution, whereas those predicted by interFoam and Fluent travel slower. The (leading) phase error using MARS2 diminishes rapidly as mesh resolution increases, confirming proper mesh convergence. Conversely, the (lag) phase error using interFoam grows progressively with mesh refinement, indicating that the predicted wave crest location increasingly diverges from its analytical counterpart. And Fluent displays a non-convergence behavior with mesh refinement in computing wave phase. MARS2 therefore appears better than interFoam and Fluent at controlling numerical dispersion and at reducing phase shift between predicted and analytical movement of the wave crest. Comparing Fluent with interFoam, the error of computing both wave height and phase shift using 3rd-order MUSCL scheme is smaller than that of 2nd-order one, which emphasizes the significance of higher order discretization during wave calculation.

Fig. 10 illustrates solitary wave free surface profiles for $H/d = 0.1$ at $t = 10$ s, with detailed zoom-ins of the wave crests provided for clarity. The waveforms are obtained by extracting the interface corresponding to the 0.5 contour of the VOF function. As shown in Fig. 10(a), the waveforms predicted by MARS2 on all three meshes exhibit satisfactory agreement with the analytical solution. Almost exact agreement is achieved between the MARS2 simulation on the finest mesh and the analytical solution, again confirming that mesh convergence has been achieved. Fig. 10(b) and Fig. 10(b) and show the corresponding results obtained by interFoam and Fluent respectively. It can be seen that larger discrepancies with the analytical waveforms occur using these two solvers than MARS2. In addition, small oscillations are observed at the trailing edge of the free surface predicted by Fluent on the coarsest mesh.

Fig. 11 displays the steady state waveforms when $H/d = 0.3$ at $t = 250$ s obtained on the coarsest $1472 \times$

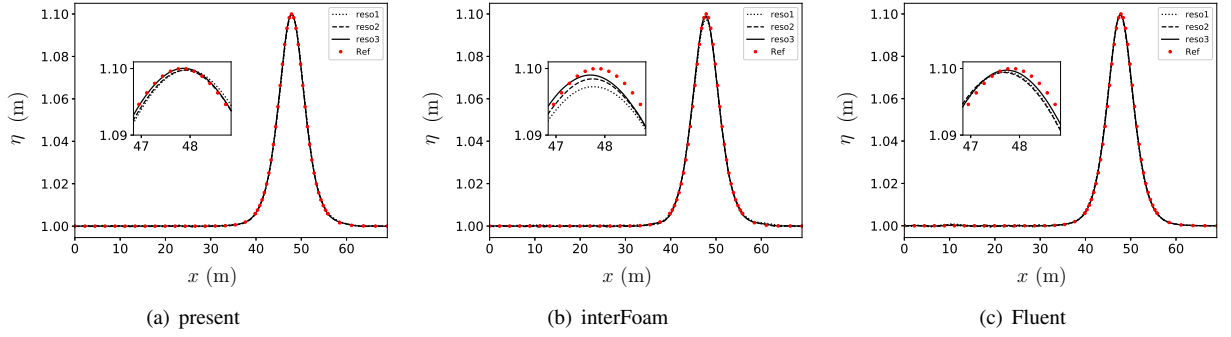(a) present   (b) interFoam   (c) Fluent

Figure 10: Solitary wave profiles for $H/d = 0.1$ at $t = 10$ s.

32 mesh (Resolution 1). It illustrates the effect of time step on relatively long duration predictions of the solitary waveform obtained by modifying the maximum Co number on the same (Resolution 1) mesh. The predictions by MARS2 (Fig. 11(a)) are much less sensitive to choice of time step than those by interFoam (Fig. 11(b)) where errors in amplitude and phase shift grow unacceptably large as the value of maximum Co deviates from about 0.1. MARS2 provides a long duration solitary wave form with acceptable amplitude and phase difference even for the largest value of maximum C_o.
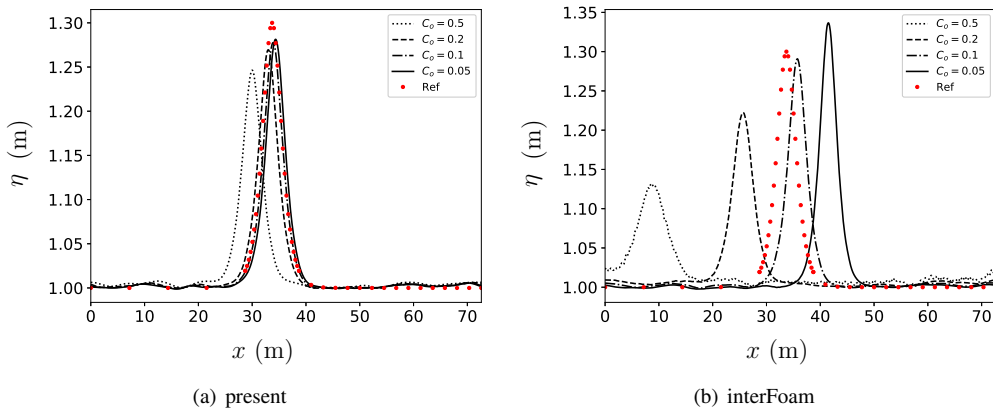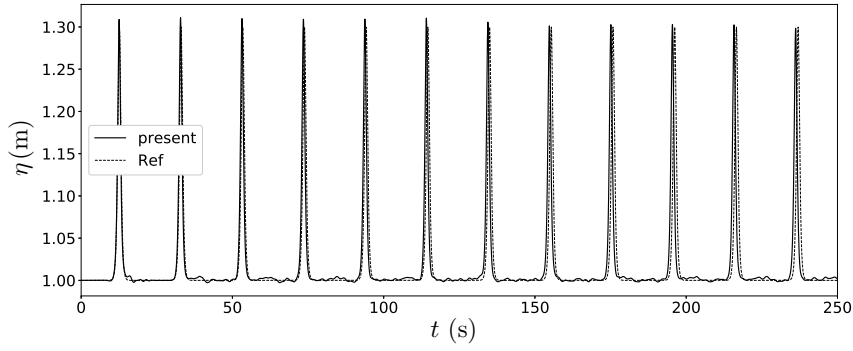

(a) present   (b) interFoam

Figure 11: Solitary wave profiles for $H/d = 0.3$ at $t = 250$ s on the 1472×32 (Resolution 1) mesh.

Fig. 12 displays the free surface profile of the solitary wave, superimposed at repeated time intervals for max $C_o = 0.2$ on the (Resolution 2) mesh. The time intervals match the propagation of the solitary wave past a virtual gauge at $x = 60$ m, which occurs repeatedly because of the periodic boundary condition employed at inlet and outlet. Once the solitary wave has passed $x = 60$ m for the first time, it will propagate through this position recurrently with a period equaling to L_x/c.

From Fig. 12 it may be discerned that both MARS2 and interFoam predictions exhibit a slight growth in phase difference with the analytical solution as the simulation progresses. In a similar fashion to the results for $H/d = 0.1$, the MARS2-simulated profile travels faster than the analytical solution, whereas the interFoam profile lags behind. Although the interFoam phase error appears later than for MARS2, the error accumulates more rapidly, revealing the cumulative effect of numerical dispersion on the simulation of a solitary wave of larger height.

Fig. 12(a) shows that the solitary wave height predicted by MARS2 initially rises slightly above the analytical value (which should remain constant at 0.3 m according to potential theory). One reason for this could be because the initial velocity field provided by Laitone's second-order theory is derived from the depth-averaged velocity obtained from shallow water theory, causing the incompressible continuity equation not to be satisfied throughout the computational domain. Fig. 13 displays the magnitude of divergence in the theoretical velocity field at $t = 0$ s, where non-zero values

(a) present
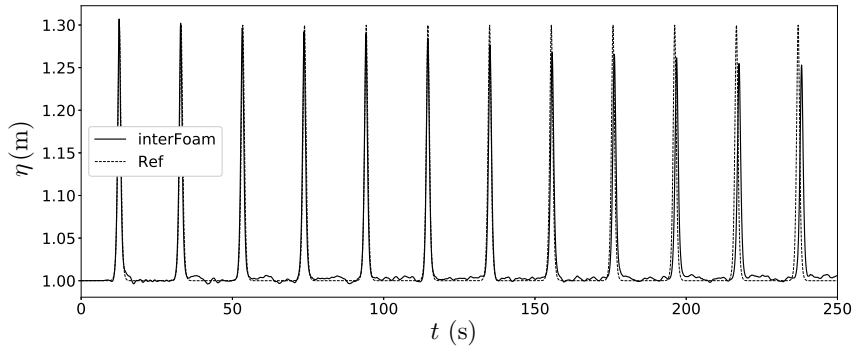
(b) interFoam

Figure 12: Evolution of solitary waveform at repeated time intervals for $H/d = 0.3$ as the crest passes $x = 60$ m on the 2944×64 (Resolution 2) mesh.

arise in the region of the solitary wave crest. Although violation of the divergence-free condition initially perturbs the flow simulation, the present model immediately resolves these in a single time step. A second, more important, reason for the growth in wave height arises from nonlinear behaviour at the larger H/d ratio, exacerbated by the present MARS2 model that uses linearized numerical methods to solve the NS equations.
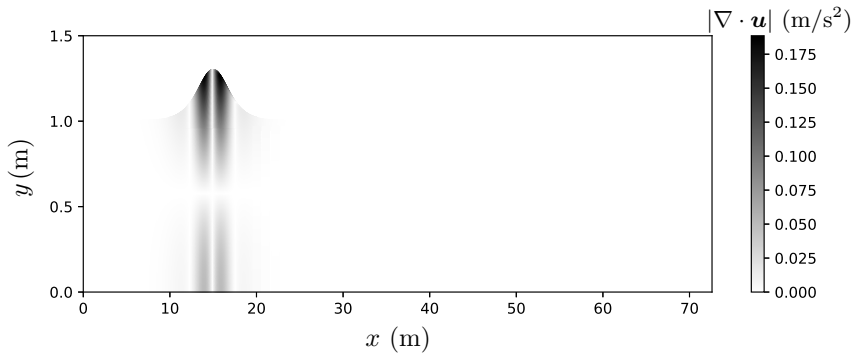

Figure 13: Theoretical magnitude of divergence in velocity field of a solitary wave with $H/d = 0.3$ at $t = 0$ s.

At later time ($t > 120\,s$), the solitary wave height amplitude predicted by MARS2 slightly decreases until stabilizing at the initial wave height without attenuation. This wave height adjustment drives the phase difference in the MARS2

simulation. By contrast, the wave height calculated by interFoam declines monotonically over time in Fig. 12(b) due to an unacceptable level of numerical dissipation.

To eliminate post-processing error, we compute the L_1 error function directly during simulation as

$$E(L_1) = \frac{\sum_{i=1}^N \left| \phi_i^{\text{Ref}} - \phi_i^{\text{Calc}} \right| V_i}{\sum_{i=1}^N \phi_i^{\text{Ref}} V_i}, \tag{17}$$

where N is the number of cells, ϕ the volume fraction and V the cell volume. $E(L_1) = 0$ for the analytical solution, and so its numerical value quantitatively reflects error in the free surface calculation.

Fig. 14 presents the temporal evolution of $E(L_1)$ for solitary wave predictions with $H/d = 0.1$ on the three meshes. It should be noted that the closer $E(L_1)$ is to zero, the more accurate are the computations. In all cases, $E(L_1)$ soars dramatically for $t < 0.2$ s and then grows almost linearly until the simulations finish at $t = 12$ s. $E(L_1)$ for MARS2 is invariably smaller than for interFoam at any time on each mesh. Again, owing to its use of high-order reconstruction schemes, the MARS2 simulations experience less numerical dissipation and dispersion effects than interFoam.
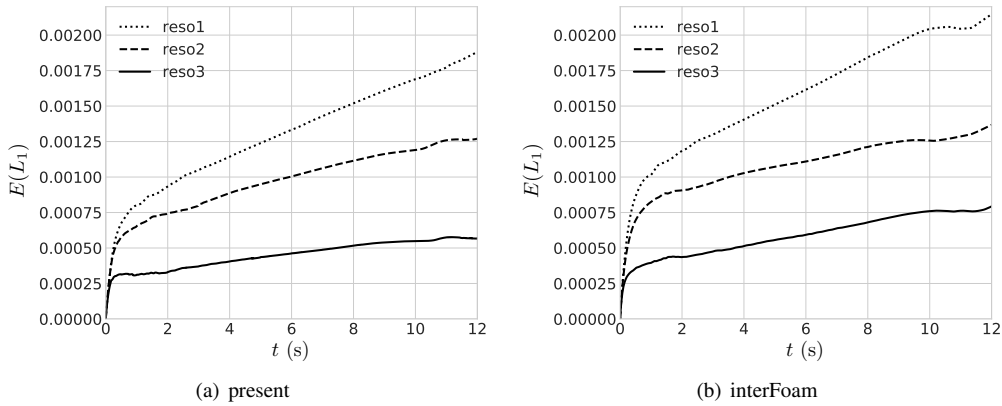

(a) present   (b) interFoam

Figure 14: Temporal evolution of $E(L_1)$ error for a solitary wave with $H/d = 0.1$.

4.1.3. Accuracy of velocity profiles

Wemmenhove, Luppes, Veldman and Bunnik (2015) reported the presence of spurious currents at the air-liquid interface induced by gravity effects in the air-liquid mixture cells used in free surface flow models. To examine whether spurious currents are generated by MARS2, we now study the detailed vertical profile of the horizontal component of fluid particle velocity at the location of the analytical wave crest for the solitary wave with $H/d = 0.1$ at time $t = 10$ s in Fig. 15.

According to Laitone's second-order theory, Eqn. (22) in Appendix A, the horizontal velocity component should progressively increase with elevation throughout the water phase. For most of the flow depth, the numerical results follow this trend, while being slightly smaller than analytical. However, in the wave region, the predicted horizontal water particle velocity component undergoes an abrupt, but small decrease, and reaches a local minimum, before shooting up to exceed the reference solution, peaking near the air-water interface. In the air phase, the horizontal fluid velocity component plummets to a negative value, before rallying to zero as the elevation increases. All simulations share a common feature in that the horizontal water particle velocity component is maximum near the free surface. The velocity overshoot corresponds to the Gibbs phenomenon at the interface where there is a fluid discontinuity. In the absence of fluid viscosity in the numerical model, there is no boundary layer present to smooth out the velocity jump at the interface (Wroniszewski et al., 2014). Fig. 15 shows that the local minimum in horizontal water particle velocity component computed by MARS2 is situated increasingly close to the wave crest as the mesh is refined, and almost vanishes on the mesh of highest resolution. Although The velocity overshoot calculated by Fluent is comparable with MARS2 in the coarser meshes, its prediction has larger discrepancies from analytical solution than MARS2 in the main streamflow. Besides, the local minimum bulge in water phase still exist in Fluent's result, indicating a lower
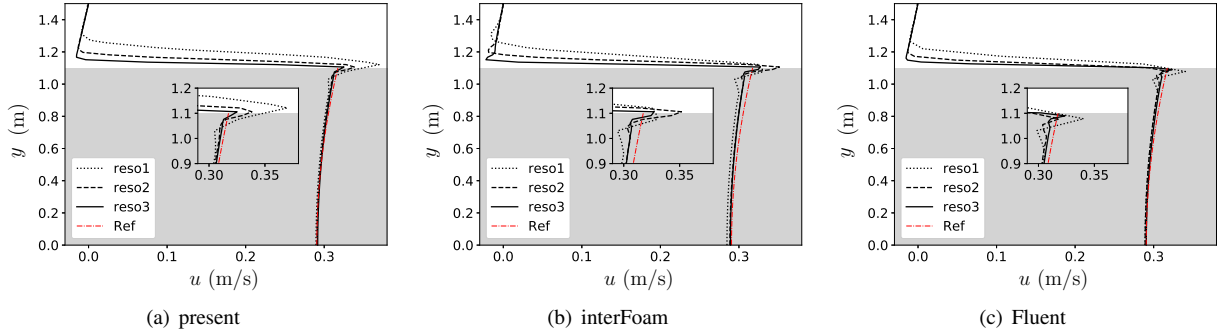
Figure 15: Vertical profiles of the horizontal fluid particle velocity component at the analytical location of a solitary wave crest for $H/d = 0.1$ at $t = 10$ s on three regular meshes. The water phase is represented by a gray background and the air phase by a white background.

convergence rate. The velocity profiles simulated using interFoam exhibit greater local changes near the free surface, except perhaps on the coarsest mesh. Unlike the interFoam predictions, there is no bulge in the horizontal air particle velocity component simulated by MARS2 and Fluent. On the other hand, a closer look at Fig. 15(b) reveals that interFoam has not yet achieved convergence in calculating the velocity near the wave crest.

4.2. Solitary wave propagation on stretched and hybrid meshes

When the above test cases were repeated on stretched and hybrid meshes, no distinct mesh-related difference could be discerned in the predictions of total energy and waveform compared to those obtained using the MARS2 and interFoam solvers. Discrepancies did occur in the horizontal velocity component profiles obtained on the stretched and hybrid meshes, as can be seen in Fig. 16. The profile obtained on the stretched mesh using interFoam displays greater fluctuations in both the water and air phases than that using MARS2. Moreover, the horizontal air particle velocity profile exhibits an enormous discrepancy on the hybrid mesh, using interFoam. These results indicate that MARS2 is better suited than interFoam at modelling the horizontal fluid particle velocity profile associated with a solitary wave.
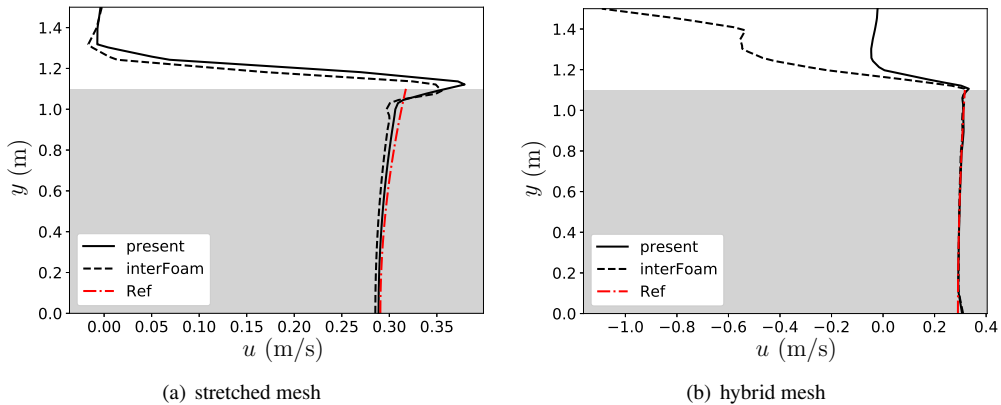

Figure 16: Vertical profiles of horizontal fluid particle velocity component at the analytical location of a solitary wave crest for $H/d = 0.1$ at $t = 10$ s on stretched and hybrid meshes. The water phase is represented by a gray background and the air phase by a white background.

To inspect the spurious counter-current obtained using the hybrid mesh in more detail, Fig. 17 displays the fluid velocity vector field obtained at $t = 10$ s using interFoam. The velocity field of the water phase predicted by interFoam is computed correctly, and is in good agreement with the MARS2 simulation. But the air velocity field predicted by interFoam appears chaotic, unlike that by MARS2. This implies that InterFoam has difficulty in simulating the air
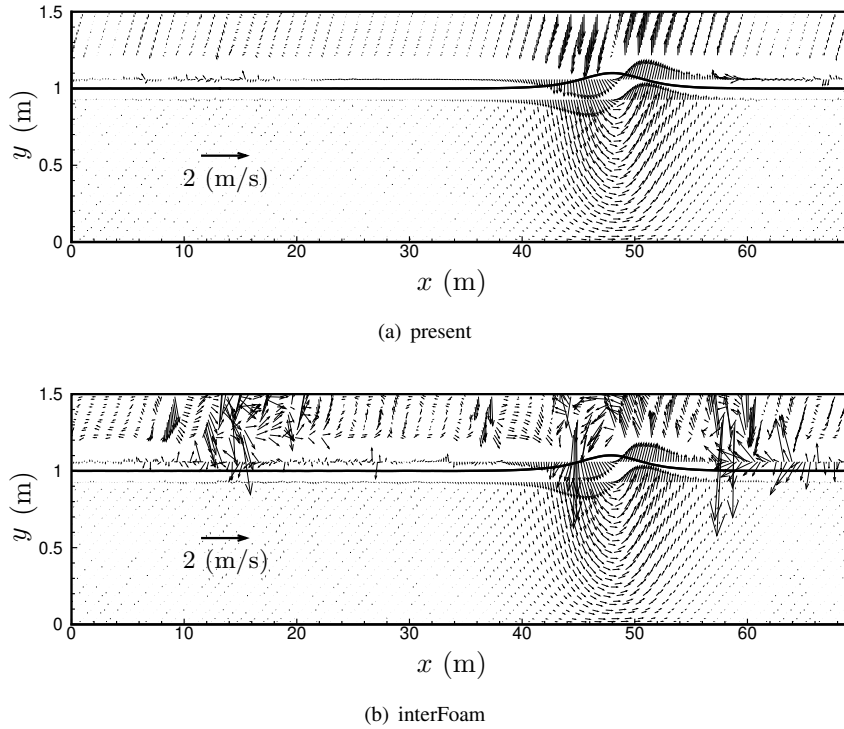
(a) present

(b) interFoam

Figure 17: Velocity vectors of a solitary wave with $H/d = 0.1$ at $t = 10$ s predicted using MARS and interFoam on hybrid meshes.

phase above solitary waves on compound unstructured meshes. In conclusion, MARS2 works well on all three types of polyhedral unstructured meshes considered.

4.3. Predictions on runup of a solitary wave

We now consider simulations by MARS2 of solitary wave runup on a plane beach, compared with interFoam. First, we study inviscid non-breaking solitary waves of relative wave amplitude $H/d = 0.1$ and 0.3, interacting with plane beaches of slope 10° and 30°. Here, the runup height is defined as the maximum elevation of the free surface above mean water level. Reference runup height to depth ratios, following Wroniszewski et al. (2014) calculated by a fully non-linear boundary integral solver, are 0.369 and 1.275 for solitary waves of $H/d = 0.1$ and 0.3 respectively on a 10° slope. For $H/d = 0.3$, the reference runup height is 0.869 on the same beach. Table 4 lists the model parameters and key nondimensionalized numerical results; it is obvious that the MARS2 predictions approach the reference runup height on the finest mesh, and outperform the corresponding results computed by interFoam and reported by Wroniszewski et al. (2014) where severe attenuation is observed.

Second, three-dimensional effects and viscosity are taken into consideration, and the results compared against a laboratory experiment by Lin et al. (1999). To overcome problems from initial transients, $t = 0$ s is consistently defined as the time when the wave peak passes $x = 7.822$ m as shown in Fig. 18.

Fig. 19 presents a series of predicted and measured free surface elevation profiles and velocity vectors at different times as the solitary wave interacts with the plane beach. Fig. 20, Fig. 21, and Fig. 22 display predicted and measured vertical profiles of horizontal water particle velocity at different times at the toe of the beach, half way up the beach, and at the still water level respectively. In general, the numerical results fit the experimental data fairly well. The crest of the solitary wave reaches the slope at $t = 0.7$ s, with maximum runup height of approximately 0.072 m attained at $t = 0.9$ s. Later, run down occurs with the nadir occurring at $t = 1.5$ s. The rundown wave then reflects, causing a secondary, much smaller runup to form at $t = 1.7$ s (Lin et al., 1999). In this case, the numerical predictions of primary runup, rundown, and secondary runup accord closely with experimental observations.

To figure out whether there exists three-dimensional effects in simulating solitary wave runup, Fig. 23 illustrates

Table 4
Solitary wave run up model parameters and predicted runup heights using MARS2 and interFoam.

$h_{\text{num}}/h_{\text{ref}}$	$H/d = 0.1$			$H/d = 0.3$		
$L_x \times L_y$	34m × 1.7m			39m × 2.6m		
MARS2	640×32	1280×64	2560×128	480×32	960×64	1920×128
10°	0.9808	0.9898	0.9987	0.9096	0.9483	0.9749
30°	–	–	–	0.9421	0.9504	0.9616
interFoam	640×32	1280×64	2560×128	480×32	960×64	1920×128
10°	0.9661	0.9730	0.9836	0.7203	0.8245	0.8627
30°	–	–	–	0.7805	0.9037	0.9444
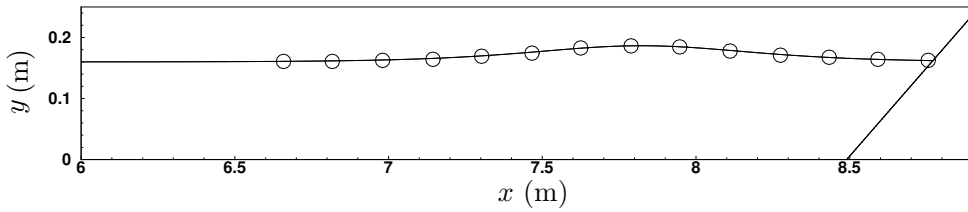
Figure 18: Free surface elevation profile of solitary wave interaction with a 30° sloping beach at $t = 0$ s and $z = 0.3$ m. (— 3D MARS2 simulation, ○ Lin et al. (1999)'s experiment.)

free surface elevation profiles predicted by 3D MARS2, 2D MARS2 and 2D interFoam at different times, with detailed zoom-ins at where solitary wave interacts with the sloping beach for clarity. Please note that the vertical coordinates in each subfigure are different and strongly zoomed in to separate the numerical results(, leading to a seemingly large deviation from the experiment). The wave profile computed by 3D MARS2 is more accurate than that by 2D MARS2, mainly reflecting in the overall decrease of surface elevation, which is promoted by three-dimensional viscosity. On the other hand, interFoam 2D prediction has the largest discrepancy from experiment's measurement. What's even worse is that strong oscillation occurs in the region of wave-beach interactions, which is distinctly non-physical.

The foregoing agreement with experimental data indicates that MARS2 has a considerable potential as a numerical tool to tackle long duration wave problems involving beaches. The present model is capable of simulating 2D and 3D wave runup and rundown for both inviscid and viscous flows, an essential prerequisite for oceanographic and coastal engineering applications.

5. Concluding remarks

A sophisticated model, called MARS2, has been constructed on polyhedral unstructured meshes for high-fidelity simulations of solitary wave propagation on plane and sloping beaches. The FVMS3 scheme and THINC/QQ method were employed respectively to solve the fluid dynamics and to capture the free surface. Initial kinematic data were generated by Laitone's second-order theory, and embedded in the present model.

For solitary wave propagation, the performance of the present model has been assessed by comparison against analytical solutions from potential theory and numerical results obtained using interFoam and Fluent. Results for inviscid solitary waves with two relative amplitudes were interpreted in terms of energy conservation, wave shape preservation, and accuracy of vertical profile of horizontal fluid particle component. It was found that predictions by the interFoam solver usually experienced severe energy loss, except for an isolated case where a spurious growth in energy occurred. Moreover, the solitary waves predicted by interFoam exhibited significant wave decay and phase shifts, attributed to intrinsic numerical dissipation and dispersion. The waveforms of interFoam were highly sensitive to the temporal and spatial resolution, with decreasing wave heights and phase shifts observed in cases involving large time steps and coarse meshes. For a more nonlinear solitary wave (with larger H/d ratio), it was found that refining the time step and meshes may even exacerbate numerical discrepancies in predictions by interFoam. By contrast, MARS2

appeared superior in controlling numerical dissipation and dispersion. Energy loss was largely reduced and waveforms were very consistent with reference solutions; only a very slight phase shift was observed after long-duration solitary wave propagation. For the vertical profile of horizontal fluid particle velocity component at the location of the crest of the analytical solitary wave, the overshoot of velocity in the interFoam simulation was much more critical than in the MARS2 simulation, promoting the Gibbs phenomenon, especially on the hybrid mesh. The incorrect calculation of the velocity field on hybrid mesh by interFoam gave rise to spurious currents in the air phase.

In the solitary wave runup tests, reference runup heights calculated using a fully non-linear boundary integral solver by Wroniszewski et al. (2014) and experimental data from Lin et al. (1999) were used as metrics. It has been demonstrated that the runup heights of the present model correctly approach the reference solution in the inviscid case. Flow field snapshots and vertical profiles of the horizontal fluid particle velocity component in viscous simulations also matched experimental data. The present model, MARS2, offers great potential for simulating free surface flows of either inviscid or viscous fluids, and could play an important role in solving wave problems encountered in oceanography and coastal engineering.

Acknowledgment

A. Solitary wave theory

To reconcile accuracy and efficiency, we adopt Laitone's second-order theory (Laitone, 1960), which guarantees sufficiently accurate solutions without loss of simplicity. The reference solution of the wave profile η is given by

$$\eta(x,t) = H \operatorname{sech}^2 X - \frac{3}{4}\left(\frac{H}{d}\right)^2 d \operatorname{sech}^2 X \times \left(1 - \operatorname{sech}^2 X\right) + d, \tag{18}$$

$$c = \sqrt{gd}\left[1 + \frac{1}{2}\frac{H}{d} - \frac{3}{20}\left(\frac{H}{d}\right)^2\right], \tag{19}$$

$$X = \sqrt{\frac{3H}{4d}}\left(1 - \frac{5}{8}\frac{H}{d}\right)\frac{x - x_0 - ct}{d}, \tag{20}$$

where d is the water depth, H the wave height, x_0 the initial position of the wave crest and c the wave celerity. The velocity field $\mathbf{u} = (u, v)$ is given by:

$$u(x,y,t) = \sqrt{gd}\left[\frac{\eta}{d} - \frac{1}{4}\left(\frac{\eta}{d}\right)^2 + \left(\frac{d}{3} - \frac{y^2}{2d}\right)\eta_{xx}\right], \tag{21}$$

$$v(x,y,t) = -\sqrt{gd}\,y\left[\left(1 - \frac{1}{2}\frac{\eta}{d}\right)\frac{\eta_x}{d} + \frac{d}{6}\eta_{xxx}\right], \tag{22}$$

The origin of the vertical coordinate, y, is the seabed. The total energy of the solitary wave is expressed as:

$$E_{Total}(t) = E_P(t) + E_K(t), \tag{23}$$

$$E_P(t) = \int_{-\infty}^{\infty}\int_{d}^{\eta(t)} \rho g(y-d)\mathrm{d}x\mathrm{d}y, \tag{24}$$

$$E_K(t) = \int_{-\infty}^{\infty}\int_{0}^{\eta(t)} \frac{1}{2}\rho\left[u(t)^2 + v(t)^2\right]\mathrm{d}x\mathrm{d}y, \tag{25}$$

where $E_P(t)$ and $E_K(t)$ represent potential and kinetic energy respectively.

B. interFoam solver

The solution procedure of the interFoam solver over one time step is summarised below.

1. Construct a predicted velocity field by solving the momentum equation.

2. Advance pressure and velocity fields through PISO loop until the tolerance for pressure-velocity system is reached.

3. Use the conservative fluxes to solve all other equations in the system.

For interface capturing, an algebraic scheme called MULES (Multidimensional Universal Limiter with Explicit Solution) is implemented to calculate the advective fluxes of the volume fraction. In order to control numerical dissipation that may smear out the moving interface, an artificial compression term is introduced to the VOF equation (1) as follows,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) + \nabla \cdot (\phi(1-\phi)U_r) = \phi \nabla \cdot \mathbf{u}, \tag{26}$$

where U_r is a velocity field that is able properly to compress the interface. Interested readers should refer to Deshpande, Anumolu and Trujillo (2012), for more detail.

C. ANSYS Fluent

For Eulerian multiphase calculations, ANSYS Fluent solves the phase momentum equations, the shared pressure, and phasic volume fraction equations in a segregated fashion. An explicit interpolation scheme called CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) (Ubbink and Issa, 1999) is specified for the VOF and Eulerian multiphase models to track the interface. For the qth phase, the VOF equation has the following form:

$$\frac{1}{\rho_{\mathrm{q}}} \left[\frac{\partial}{\partial t} \left(\phi_q \rho_q \right) + \nabla \cdot \left(\phi_q \rho_q \mathbf{u}_q \right) = \sum_{p=1}^{n} \left(\dot{m}_{\mathrm{pq}} - \dot{m}_{\mathrm{qp}} \right) \right], \tag{27}$$

where $\dot{m}_{\mathrm{qp}}$ is the mass transfer from phase q to phase p and $\dot{m}_{\mathrm{pq}}$ is the mass transfer from phase p to phase q.

Third-Order MUSCL Scheme, conceived from the original MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) by blending a central differencing scheme and second-order upwind scheme, is designate to calculate the convection term in momentum equation. For the pressure-velocity coupling, Fluent uses the phase coupled SIMPLE (PC-SIMPLE) algorithm. A pressure correction equation is built based on total volume continuity, with pressure and velocities corrected afterwards so as to satisfy the continuity constraint.

References

Abadie, S., Morichon, D., Grilli, S., Glockner, S., 2010. Numerical simulation of waves generated by landslides using a multiple-fluid navier–stokes model. Coastal engineering 57, 779–794.

Ablowitz, M.J., Kruskal, M.D., Ladik, J., 1979. Solitary wave collisions. SIAM Journal on Applied Mathematics 36, 428–437.

Aulisa, E., Manservisi, S., Scardovelli, R., Zaleski, S., 2003. A geometrical area-preserving volume-of-fluid advection method. Journal of Computational Physics 192, 355–364.

Bona, J.L., Pritchard, W., Scott, L.R., 1980. Solitary-wave interaction. The Physics of Fluids 23, 438–441.

Chorin, A.J., 1968. Numerical solution of the navier-stokes equations. Mathematics of computation 22, 745–762.

Deshpande, S.S., Anumolu, L., Trujillo, M.F., 2012. Evaluating the performance of the two-phase flow solver interfoam. Computational Science & Discovery 5, 014016.

Formaggia, L., Miglio, E., Mola, A., Parolini, N., 2008. Fluid–structure interaction problems in free surface flows: application to boat dynamics. International Journal for Numerical Methods in Fluids 56, 965–978.

Harlow, F.H., Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. The physics of fluids 8, 2182–2189.

Harten, A., 1978. The artificial compression method for computation of shocks and contact discontinuities. iii. self-adjusting hybrid schemes. Mathematics of Computation 32, 363–389.

Kim, J., Moin, P., 1985. Application of a fractional-step method to incompressible navier-stokes equations. Journal of computational physics 59, 308–323.

Laitone, E., 1960. The second approximation to cnoidal and solitary waves. Journal of fluid mechanics 9, 430–444.

Leng, Y., Li, X., Fu, D., Ma, Y., 2012. Optimization of the muscl scheme by dispersion and dissipation. Science China Physics, Mechanics and Astronomy 55, 844–853.

Lin, P., Chang, K.A., Liu, P.L.F., 1999. Runup and rundown of solitary waves on sloping beaches. Journal of waterway, port, coastal, and ocean engineering 125, 247–255.

Liu, J., Gollub, J., 1994. Solitary wave dynamics of film flows. Physics of Fluids 6, 1702–1712.

Lo, E.Y., Shao, S., 2002. Simulation of near-shore solitary wave mechanics by an incompressible sph method. Applied Ocean Research 24, 275–286.

Lubin, P., 2004. Simulation des grandes échelles du déferlement plongeant des vagues. Ph.D. thesis. Bordeaux 1.

Maxworthy, T., 1976. Experiments on collisions between solitary waves. Journal of Fluid Mechanics 76, 177–186.

Montagna, F., Bellotti, G., Di Risio, M., 2011. 3d numerical modeling of landslide-generated tsunamis around a conical island. Natural hazards 58, 591–608.

Munk, W.H., 1949. The solitary wave theory and its application to surf problems. Annals of the New York Academy of Sciences 51, 376–424.

Park, J.S., Kim, C., 2010. Multi-dimensional limiting process and maximum principle for the computations of hyperbolic conservation laws on unstructured grids .

Parker, B., Youngs, D., 1992. Two and three dimensional Eulerian simulation of fluid flow with material interfaces. Atomic Weapons Establishment.

Perić, R., Abdel-Maksoud, M., 2015. Generation of free-surface waves by localized source terms in the continuity equation. Ocean Engineering 109, 567–579.

Shu, C.W., Osher, S., 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes. Journal of Computational Physics 77, 439–471.

Tryggvason, G., Scardovelli, R., Zaleski, S., 2011. Direct Numerical Simulations of Gas–Liquid Multiphase Flows. Cambridge University Press.

Ubbink, O., Issa, R., 1999. A method for capturing sharp fluid interfaces on arbitrary meshes. Journal of Computational Physics 153, 26–50.

Wemmenhove, R., Luppes, R., Veldman, A.E., Bunnik, T., 2015. Numerical simulation of hydrodynamic wave loading by a compressible two-phase flow method. Computers & Fluids 114, 218–231.

Wroniszewski, P.A., Verschaeve, J.C., Pedersen, G.K., 2014. Benchmarking of navier–stokes codes for free surface simulations by means of a solitary wave. Coastal Engineering 91, 1–17.

Xiao, F., Ikebata, A., 2003. An efficient method for capturing free boundaries in multi-fluid simulations. International Journal for Numerical Methods in Fluids 42, 187–210.

Xie, B., Deng, X., Liao, S., 2019. High-fidelity solver on polyhedral unstructured grids for low-mach number compressible viscous flow. Computer Methods in Applied Mechanics and Engineering 357, 112584.

Xie, B., Jin, P., Du, Y., Liao, S., 2020. A consistent and balanced-force model for incompressible multiphase flows on polyhedral unstructured grids. International Journal of Multiphase Flow 122, 103125.

Xie, B., Xiao, F., 2017. Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids: The thinc method with quadratic surface representation and gaussian quadrature. Journal of Computational Physics 349, 415–440.

Youngs, D.L., 1982. Time-dependent multi-material flow with large fluid distortion. Numerical methods for fluid dynamics .

Zhang, Z., Zhao, X., Xie, B., Nie, L., 2019. High-fidelity simulation of regular waves based on multi-moment finite volume formulation and thinc method. Applied Ocean Research 87, 81–94.
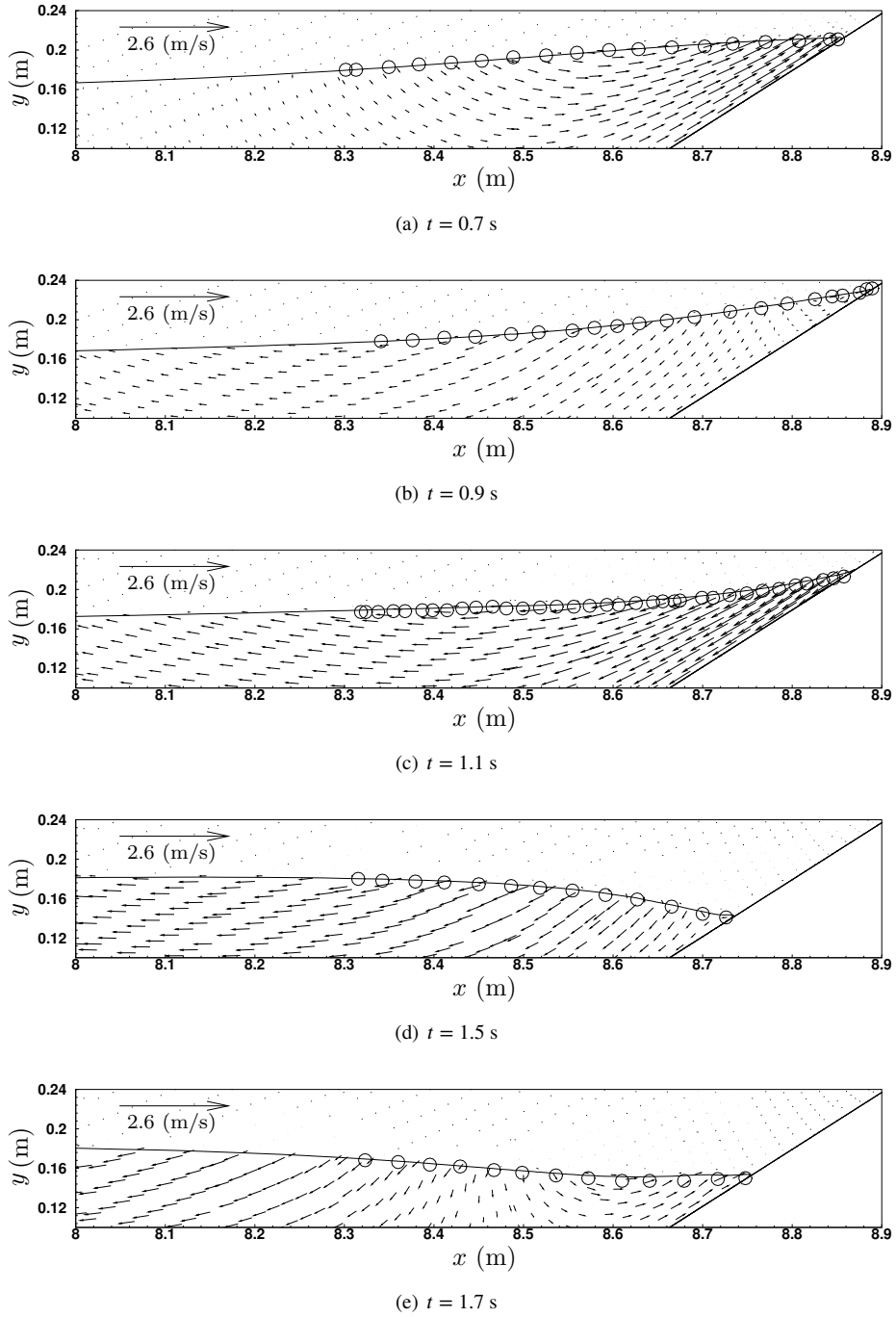
(a) $t = 0.7$ s

(b) $t = 0.9$ s

(c) $t = 1.1$ s

(d) $t = 1.5$ s

(e) $t = 1.7$ s

Figure 19: Free surface elevation profiles and velocity vectors of solitary wave runup and rundown on a 30° slope at $t = 0.7$, 0.9, 1.1, 1.5 and 1.7 s. (— 3D MARS2 simulation, ○ Lin et al. (1999)'s experiment.)
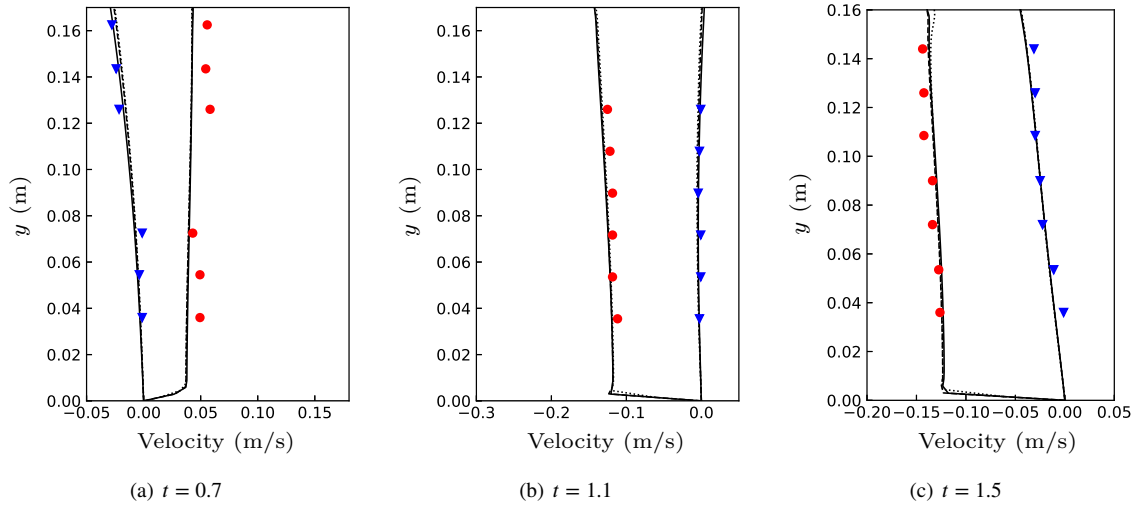
Figure 20: Vertical profiles of water particle velocity components for solitary wave runup and rundown at $x = 8.3972$ m. (— : u and v predicted by 3D MARS2, - -: by 2D MARS2, ⋯: by 2D interFoam, • & ▼ measured by Lin et al. (1999).)
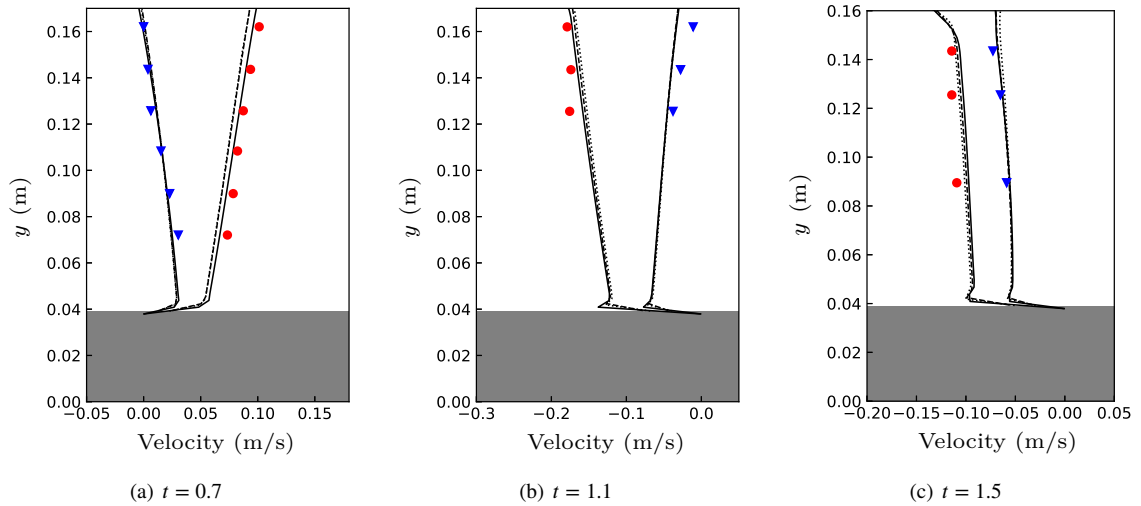

Figure 21: Vertical profiles of water particle velocity components for solitary wave runup and rundown at $x = 8.5556$ m. (— : u and v predicted by 3D MARS2, - -: by 2D MARS2, ⋯: by 2D interFoam, • & ▼ measured by Lin et al. (1999). The solid slope is represented with gray background and the water phase with white background.)
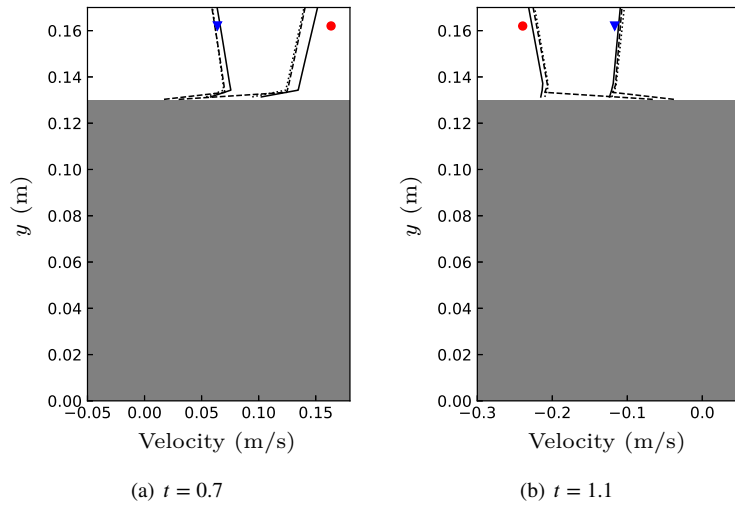
(a) $t = 0.7$ (b) $t = 1.1$

Figure 22: Vertical profiles of water particle velocity components for solitary wave runup and rundown at $x = 8.7146$ m. (— : u and v predicted by 3D MARS2, - -: by 2D MARS2, ⋯: by 2D interFoam, • & ▼ measured by Lin et al. (1999). The solid slope is represented with gray background and the water phase with white background.)
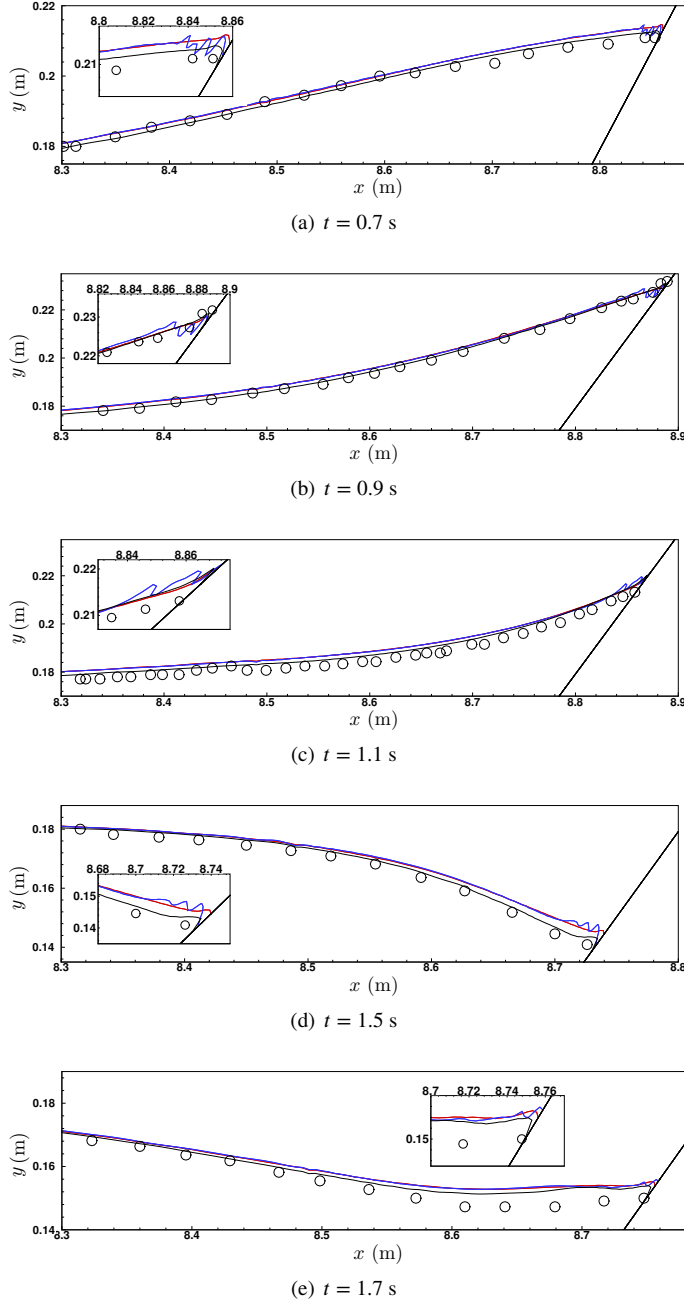
(a) $t = 0.7$ s

(b) $t = 0.9$ s

(c) $t = 1.1$ s

(d) $t = 1.5$ s

(e) $t = 1.7$ s

Figure 23: Free surface elevation profiles of solitary wave runup and rundown on a 30° slope at $t = 0.7$, 0.9, 1.1, 1.5 and 1.7 s. (black line: predicted by 3D MARS2, red line: by 2D MARS2, blue line: by 2D interFoam, ∘: by Lin et al. (1999)'s experiment.)