Finite volume arbitrary Lagrangian-Eulerian schemes using dual meshes for ocean wave applications

Martin Ferrand, Jeffrey C Harris

Highlights

Finite volume Arbitrary Lagrangian-Eulerian schemes using dual meshes for ocean wave applications

Martin Ferrand,Jeffrey C. Harris

- Finite volume three-dimensional Navier-Stokes modeling of water wave propagation
- Steep wave generation and propagation with Arbitrary Lagrangian-Eulerian scheme
- Use of Compatible Discrete Operators shows improved accuracy and stability

Finite volume Arbitrary Lagrangian-Eulerian schemes using dual meshes for ocean wave applications

Martin Ferrand[a], Jeffrey C. Harris[b,*]

[a]CEREA, Ecole des Ponts, EDF R et D, Chatou, France
[b]LHSV, Ecole des Ponts, CEREMA, EDF R et D, Chatou, France

ARTICLE INFO

Keywords:
Code_Saturne
Navier-Stokes model
Wave propagation
Arbitrary Lagrangian-Eulerian

ABSTRACT

For reasons of efficiency and accuracy, water wave propagation is often simulated with potential or inviscid models rather than Navier-Stokes solvers, but for wave-induced flows, such as wave-structure interaction, viscous effects are important under certain conditions. Alternatively, general purpose Navier-Stokes (CFD) models can have limitations when applied to such free-surface problems when dealing with large amplitude waves, run-up, or propagation over long distances. Here we present an Arbitrary Lagrangian-Eulerian (ALE) algorithm with special care to the time-stepping and boundary conditions used for the free-surfaces, integrated into *Code_Saturne*, and we test its capabilities for modeling a variety of water wave generation and propagation benchmarks, and finally consider interaction with a vertical cylinder. Two variants of the mesh displacement computation are proposed and tested against the discrete Geometric Conservation Law (GCL). The more robust variant, for highly curved or sawtoothed free-surfaces, uses a Compatible Discrete Operator scheme on the dual mesh for solving the mesh displacement, which makes the algorithm valid for any polyhedral mesh. Results for standard wave propagation benchmarks for both variants show that, when care is taken to avoid grids with excessive numerical dissipation, this approach is effective at reproducing wave profiles as well as forces on bodies.

1. Introduction

There are many applications to models of water waves, from understanding tsunami hazards, to computing the forces on offshore structures, to understanding the performance of different ship designs. Similarly, there are many different methods for modeling water waves; a recent summary was given by [35]. Because of the vast horizontal scales required to study ocean waves, often a simplified approach is used, such as potential flow, with either nonlinear or linearized free-surface boundary condition, or a depth-averaged approach, such as shallow water or Boussinesq equations. Such specialized models efficiently propagate waves over long distances without excessive numerical dissipation.

A direct approach to the physics would be to solve the full three-dimensional (3-D) Navier-Stokes equations, in a domain covering both the water and air phases. In fact, some of the first water wave models solved the Navier-Stokes equations in this way [12], using the Marker And Cell (MAC) method [26]. Many different methods for tracking the free-surface have been developed, but variations of the Volume-Of-Fluid (VOF) technique [30] have become quite popular for coastal engineering, using it for wave modeling [*e.g.*, 33] and wave-structure interaction [*e.g.*, 14], as well as many others applications. Such approaches have the advantage of being able to treat breaking waves, which is an important but complex issue ([*e.g.*, 27]). One major drawback is they require finer computational domains and tracking the interface without diffusion is a challenge. For a recent discussion of wave propagation benchmarks for VOF codes, see [51].

Alternatively, one can use a boundary-fitted domain, and move the computational domain at each time-step [29]. Such an arbitrary Lagrangian-Eulerian (ALE) approach will be considered here. This is used in many ocean models [*e.g.*, Telemac3D; 28], or non-hydrostatic wave models [38], and these are able to solve such diverse applications as landslide-generated tsunamis and surf-zone wave propagation. Many models that use a boundary-fitted domain, however, are based on sigma-coordinate grids in the vertical, which can have some limitations for dealing with complex grids or immersed bodies. These limitations are removed with the present ap-

*Corresponding author
martin.ferrand@edf.fr (M. Ferrand)
ORCID(s): 0000-0002-3995-1503 (M. Ferrand); 0000-0002-8167-7974 (J.C. Harris)

proach, where a more general mesh movement is solved with specific free-surface condition for the mesh displacement. Although ALE cannot model the details of overturning waves like a VOF approach, such non-hydrostatic wave models are able to incorporate energy dissipation in order to represent breaking effects [53], and examples of this will be shown below.

In the context of ALE solvers, there have been many developments (see e.g., [17]). The sigma-coordinate grid mentioned above corresponds to a special case of the simplest mesh regularization approach, that of a transfinite mapping. To apply to more general meshes, a common approach that will be discussed further below is a Laplacian smoothing approach (e.g., Benson [6]) where a Laplace or Poisson equation is solved for the mesh velocity in the interior of the domain. One can consider other node solvers, and there exist more complex, high-order shock-capturing schemes (e.g., [8]). As well, general ALE solvers often consider adaptive meshes, with a remeshing of the domain during the simulation, in order to handle large mesh displacements (e.g., [43]). Such developments however can be time-consuming (due to additional equations to solve), and are unnecessary for ocean wave problems, as we consider waves that are not overturning. As a result, we consider only the essential aspects of integrating an ALE scheme into an existing Navier-Stokes solver and the resulting performance for ocean wave applications. For this, it is important to consider the discrete Geometric Conservation Law (GCL), which states that the volume change of a cell during a timestep should be equal to the volume swept by the cell boundaries during that timestep. The GCL is critical to ensure that the algorithm is robust (e.g., conserves mass and energy) for long time simulations [*e.g.* 48, 52, 21, 45].

As a result, we consider below the numerical details required to develop a robust ALE finite volume algorithm integrated in the open-source CFD solver *Code_Saturne* [2] for version 6.1 and newer, with particular attention to the discrete GCL. *Code_Saturne* uses collocated finite volumes to solve the Navier-Stokes, Reynolds Averaged Navier-Stokes (RANS), or Large Eddy Simulation (LES) equations, and has been used extensively in the industry, from nuclear thermal hydraulic application [see 24, for a recent application], atmospheric modeling [*e.g* 3], electric arc modeling [*e.g* 13], fire modeling [39], ventilation, combustion (*e.g.*, coal, gas, biomass, [16]) and has also been used previously for wave-current interaction [*e.g.* 47, 5].

The paper is organized as follows. Before passing to the specific case of incompressible flows of constant density used in all applications here, in section 2, we present the governing equations of the finite volume space and time discretization for fixed meshes in terms that are valid for both incompressible or compressible flows. Then in section 3 we present the ALE algorithm and specify the free-surface boundary condition, with special care on the time stepping scheme to conserve uniform velocity fields. The original use of cell-based, for fluid quantities, and vertex-based solvers (on the dual barycentric mesh), for the mesh displacement, is also presented, and this improves on volume conservation. In order to facilitate very steep waves, a filtering of spurious waves of length smaller than the mesh resolution is also introduced. Finally, in section 4, we consider numerous validation cases. This is first done with flow through a valve, forced by a piston, showing that the velocity field remains accurate with a moving mesh, then by looking at standing waves of a range of amplitudes to verify volume conservation and model stability, a critical point for moving mesh algorithms. Then, satisfied that the model is robust, we consider a number of different applications, including vis-

cous damping of standing waves, solitary wave propagation, wave propagation over a submerged bar, wave forces on a monopile, and the modeling of a novel laboratory setup for long wave generation. This ensemble of test cases demonstrates the ability of this approach to handle a wide variety of test cases of interest to the ocean modeling community.

2. Methodology

2.1. Governing equations

To describe the governing equations, we begin with the conservative Cauchy momentum equations written in vector notation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \underline{u}) &= \Gamma, \\ \frac{\partial}{\partial t}(\rho \underline{u}) + \underline{\operatorname{div}}(\underline{u} \otimes \rho\, \underline{u}) &= \underline{\operatorname{div}}(\underline{\underline{\sigma}}) + \underline{S}_u + \Gamma \underline{u}^{in}, \end{aligned} \tag{1}$$

where $\underline{u}$ represents the fluid velocity, ρ is the density of the fluid, $\underline{\underline{\sigma}}$ is the stress field, $\underline{S}_u$ is a field of source terms (*e.g.*, gravity, Coriolis, additional head losses), Γ a possible mass source, and $\underline{u}^{in}$ the velocity of a possible injected fluid. For sake of simplicity, $\underline{S}_u$ and Γ will be set to zero hereafter.

The stress tensor $\underline{\underline{\sigma}}$ is decomposed into an isotropic component, called the pressure p, and the deviatoric component, $\underline{\underline{\tau}}$:

$$\underline{\underline{\sigma}} = -p\underline{\underline{1}} + \underline{\underline{\tau}}, \tag{2}$$

and thus the Cauchy equations are:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \underline{u}) &= 0, \\ \frac{\partial}{\partial t}(\rho \underline{u}) + \underline{\operatorname{div}}(\underline{u} \otimes \rho\, \underline{u}) &= -\underline{\nabla} p + \underline{\operatorname{div}}\left(\underline{\underline{\tau}}\right). \end{aligned} \tag{3}$$

For a Newtonian fluid, the constitutive law connecting the stress tensor to the deviatoric part of the deformation rate tensor, given as $\underline{\underline{S}} \equiv \frac{1}{2}\left(\underline{\underline{\nabla}}\,\underline{u} + \underline{\underline{\nabla}}\,\underline{u}^T\right)$, is determined as:

$$\underline{\underline{\tau}}(\underline{u}) = \mu\left(\underline{\underline{\nabla}}\,\underline{u} + \underline{\underline{\nabla}}\,\underline{u}^T\right) + \left(\kappa - \frac{2}{3}\mu\right)\operatorname{tr}\left(\underline{\underline{\nabla}}\,\underline{u}\right)\underline{\underline{1}}, \tag{4}$$

where κ is the volume viscosity, usually neglected and therefore omitted hereafter in this article.

The combination of terms in the Cauchy momentum equations, Eqs. (1), and a Newtonian fluid, Eq. (4), gives the Navier-Stokes equations.

When ρ is considered constant, flow is incompressible and the mathematical model is complete, otherwise the system should be supplemented with an energy equation and an Equation Of State (EOS) between density, pressure and energy must be specified [see *e.g* 15]. Turbulent flows can also be considered subtracting the Reynolds stresses (for the RANS approach) or the sub-grid stresses (for the LES approach) from the Cauchy stresses. In this article, only incompressible constant density flows will be considered.

2.2. Space and time discretization

In this section, the time discretization is detailed on a motionless meshed domain, and some insight into the space discretization is given. A finite volume scheme where the velocity and pressure are stored at the cell centers, $\underline{x}_c$ (see Fig. 1), is used.

Integrating conservative Eqs. (3) both in space and time leads to the definition of the extensive quantities:

$$\begin{aligned} \Omega_c &= \int_{\Omega_c} \mathrm{d}\Omega, \\ M_c &= \int_{\Omega_c} \rho \mathrm{d}\Omega, \\ \underline{Q}_c &= \int_{\Omega_c} \rho \underline{u} \mathrm{d}\Omega, \end{aligned} \tag{5}$$

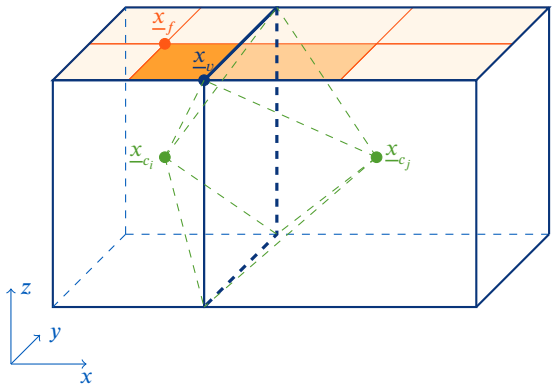


Figure 1: Labeling of information used for a mesh. In green, $\underline{x}_{c_i}$ is the barycenter of one cell c_i, whereas $\underline{x}_{c_j}$ is the one of c_j. Each face f is split into portions of faces associated to a vertex. The portion of face f associated to a vertex v is drawn in solid orange, and is defined by $\underline{x}_f$, the barycenter of f, the middle of the two edges of f passing through vertex v, and $\underline{x}_v$ itself. The dual barycentric cell around a vertex v is defined by connecting surrounding cell centers, face centers and edge centers.

for cell volume Ω_c, cell mass M_c, and cell momentum $\underline{Q}_c$. As for the intensive quantities, let ρ_c be considered the cell-averaged density, and let $\underline{u}_c$ be the density-weighted cell-averaged velocity, defined as:

$$\begin{aligned} \rho_c &= \frac{M_c}{\Omega_c}, \\ \underline{u}_c &= \frac{\underline{Q}_c}{M_c}. \end{aligned} \tag{6}$$

We denote the exponent $(.)^n$ as the state at time-step t^n, and similarly $(.)^{n+1}$ as the state at time t^{n+1}, which is separated from t^n by a time interval Δt. Time-integrals over Δt are denoted by $(.)|_n^{n+1} \equiv \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} (.)\mathrm{d}t$.

Integrating the mass equation of Eq. (1) over cell c and time interval Δt and using Gauss' theorem gives:

$$M_c^{n+1} - M_c^n + \Delta t \sum_{f\in\mathcal{F}_c} \left(\rho\,\underline{u}\right)_f \Big|_n^{n+1} \cdot \underline{S}_f = 0, \tag{7}$$

where $\mathcal{F}_c$ is the ensemble of all the planar polygonal faces of cell c and $\underline{S}_f$ is the outward surface vector. Note that the quantity $(.)_f$ coming from the integration of the divergence is defined as the average (in space) of $(.)$ over the face: $(.)_f \equiv \frac{1}{S_f}\int_f (.)\mathrm{d}S$, and its expression with respect to the cell values will be given by the space discretization.

Performing the same computation for the momentum equation of Eq. (1) gives:

$$\underline{Q}_c^{n+1} - \underline{Q}_c^n + \Delta t \sum_{f\in\mathcal{F}_c} \left(\underline{u}\otimes\rho\,\underline{u} - \underline{\underline{\sigma}}\right)_f \Big|_n^{n+1} \cdot \underline{S}_f = \underline{0}. \tag{8}$$

There are many schemes for solving the Navier-Stokes equations with the finite volume method [49]. Here, the Navier-Stokes equations are solved in two steps: a prediction step for the velocities from the explicit pressure field, and a correction step, where the continuity equation is used to calculate a change of pressure. The correction step adjusts the mass flux to ensure mass conservation and then update the velocity field. Sub-iterations for these two steps is also possible to perform a Pressure-Implicit with Splitting of Operators (PISO)-like algorithm [32]. These iterations will be denoted below by the superscript k, starting at 1.

For the prediction step, by default, an implicit Euler first-order scheme is used, where the viscous shear stress is time-implicit, the pressure is time-explicit, and the convective term is linearized and semi-implicit as follows:

$$\begin{aligned} &\Omega_c\left(\rho_c^{n+1,k-1}\tilde{\underline{u}}_c^k - \rho_c^n\underline{u}_c^n\right) \\ &\quad + \Delta t \sum_{f\in\mathcal{F}_c} \tilde{\underline{u}}_f^k \left(\left(\rho\,\underline{u}\right)_f \Big|_n^{n+1,k-1} \cdot \underline{S}_f\right) \\ &\quad - \Delta t \sum_{f\in\mathcal{F}_c} \left(\underline{\underline{\tau}}\left(\tilde{\underline{u}}^k\right)\right)_f \cdot \underline{S}_f \\ &\quad = -\Delta t \sum_{f\in\mathcal{F}_c} \left(p_f\Big|_n^{n+1,k-1}\right)\underline{S}_f, \end{aligned} \tag{9}$$

where $\underline{\widetilde{u}}^k$ is the predicted velocity field to be solved, and the mass flux $\left(\rho\,\underline{u}\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f$ is initialized by $\left(\rho\,\underline{u}\right)_f\Big|_n^{n+1,0}\cdot\underline{S}_f = \left(\rho\,\underline{u}\right)_f\Big|_{n-1}^{n}\cdot\underline{S}_f$, $p|_n^{n+1,0} = p|_{n-1}^n$, $\rho^{n+1,0} = \rho^n$. Different convective schemes for $\underline{\widetilde{u}}_f^k$ have been implemented, including a pure upwind scheme, a Second-Order Linear Upwind (SOLU), or a centred scheme with an optional slope test limiter. This last approach is the one used for all test cases presented in this paper. The transpose of the velocity gradient and the isotropic term are in fact implicit in time, with an iterative process used to reconstruct terms due to the non-orthogonalities of the mesh if present [2].

The unsteady term of Eq. (9) is rewritten using the conservation of mass:

$$\Omega_c\left(\rho_c^{n+1,k-1}\underline{\widetilde{u}}_c^k-\rho_c^n\underline{u}_c^n\right) = \Omega_c\rho_c^n\left(\underline{\widetilde{u}}_c^k-\underline{u}_c^n\right) - \Delta t\sum_{f\in\mathcal{F}_c}\underline{\widetilde{u}}_c^k\left(\left(\rho\,\underline{u}\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f\right). \tag{10}$$

(As mentioned above, although the density is taken to be constant in this paper, the density could be updated through an equation of state, depending on scalars such as temperature or salinity; if so, this modification would be applied here, before the correction step.)

During the correction step, a simplified momentum equation is solved under the mass balance constraint:

$$\begin{aligned}\rho^{n+1,k-1}\frac{\underline{u}^{n+1,k}-\underline{\widetilde{u}}^k}{\Delta t}+\underline{\nabla}\phi^k=\underline{0},\\ \left(\rho\underline{u}\right)_f\Big|_n^{n+1,k}\cdot\underline{S}_f=\left(\rho^{n+1,k-1}\underline{u}^{n+1,k}\right)_f\cdot\underline{S}_f,\\ \Omega_c\left(\rho_c^{n+1,k}-\rho_c^n\right)+\Delta t\sum_{f\in\mathcal{F}_c}\left(\left(\rho\underline{u}\right)_f\Big|_n^{n+1,k}\right)\cdot\underline{S}_f=0,\end{aligned} \tag{11}$$

where $\phi^k = p|_n^{n+1,k} - p|_n^{n+1,k-1}$ is the pressure increment. From Eq. (11), a Poisson equation is written for ϕ^k:

$$-\sum_{f\in\mathcal{F}_c}\left(\Delta t\underline{\nabla}\phi^k\right)_f\cdot\underline{S}_f=-\sum_{f\in\mathcal{F}_c}\left(\rho^{n+1,k-1}\underline{\widetilde{u}}^k\right)_f\cdot\underline{S}_f -\Omega_c\frac{\rho_c^{n+1,k}-\rho_c^n}{\Delta t}. \tag{12}$$

A Two-Point Flux Approximation (TPFA) with non-orthogonal reconstruction is used to discretize the Laplace operator on ϕ^k [20], and the Rhie and Chow filter can be added to the right hand side of Eq. (12) [2]. See [22] for more details about the reconstruction process. Note that the first line of Eq. (11) is then used to update the cell velocity field $\underline{u}_c^{n+1,k}$.

3. ALE framework and free surface condition

Next we consider the ALE formalism, and then present the time-stepping scheme used, showing that volume is conserved when considering mesh displacement in a single dimension.

3.1. Governing equations

First it is necessary to formulate the equations governing the conservation of momentum in the domain, $\Omega(t)$, which moves in time. We denote the velocity of mesh displacement field $\underline{v}\left(\underline{x},t\right)$, for points $\underline{x}\left(\underline{x}_0,\,t\right)$ originally located at $\underline{x}_0$ and moving in the domain. The mesh velocity of $\Omega(t)$ is thus written:

$$\underline{v}\left(\underline{x},t\right)=\frac{\partial\underline{x}\left(\underline{x}_0,\,t\right)}{\partial t}. \tag{13}$$

We now consider a sub-domain $\Omega_c(t)$ of $\Omega(t)$. This domain corresponds to a cell of the mesh previously described. Using the Leibniz integration rule (or Reynolds transport theorem), we can evaluate the variation of an arbitrary ten-

sor field $\underline{v}$ in $\Omega_c(t)$ over time (here noted as a tensor of order one, *i.e.*, a vector, but the Leibniz integration rule is true for any order). This is written as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \underline{v}\mathrm{d}\Omega_c = \int_{\Omega_c(t)} \frac{\partial \underline{v}}{\partial t}\mathrm{d}\Omega_c + \int_{\partial\Omega_c(t)} \underline{v}\otimes\underline{v}\cdot\underline{\mathrm{d}S}. \quad (14)$$

It is important to note that $\frac{\mathrm{d}}{\mathrm{d}t}$ represents the total derivative of the integral quantity, that follows the movement of $\Omega_c(t)$.

Applying the Leibniz integral rule, Eq. (14), to the unity scalar field gives the way cell volumes evolve:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \mathrm{d}\Omega = \int_{\partial\Omega_c(t)} \underline{v}\cdot\underline{\mathrm{d}S}. \quad (15)$$

This last equation (in a form integrated in time) is often called the Geometrical Conservation Law (GCL) and will be described in more detail in §3.2.

Performing the same calculation for the variation of mass $\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \rho\mathrm{d}\Omega$ in the cell c and using the first line of Eq. (1) we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \rho\mathrm{d}\Omega + \int_{\partial\Omega_c(t)} \rho\left(\underline{u}-\underline{v}\right)\cdot\underline{\mathrm{d}S} = 0. \quad (16)$$

Applying again the Leibniz integral rule (14) to establish the conservation of momentum in the domain $\Omega_c(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \rho\underline{u}\mathrm{d}\Omega_c = \int_{\Omega_c(t)} \frac{\partial \rho\underline{u}}{\partial t}\mathrm{d}\Omega_c + \int_{\partial\Omega_c(t)} \rho\underline{u}\otimes\underline{v}\cdot\underline{\mathrm{d}S}. \quad (17)$$

We apply the Cauchy momentum equation to replace $\dfrac{\partial \rho\underline{u}}{\partial t}$ of Eq. (17), which gives us:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} \rho\underline{u}\mathrm{d}\Omega + \int_{\partial\Omega_c(t)} \underline{u}\otimes\rho\left(\underline{u}-\underline{v}\right)\cdot\underline{\mathrm{d}S} = \int_{\partial\Omega_c(t)} \underline{\underline{\sigma}}\cdot\underline{\mathrm{d}S}. \quad (18)$$

3.2. Space and time discretization

Let us start by integrating the volume conservation Eq. (15) over a time step Δt:

$$\Omega_c^{n+1} - \Omega_c^n = \int_{t^n}^{t^{n+1}} \left(\sum_{f\in\mathcal{F}_c} \int_f \underline{v}\cdot\underline{\mathrm{d}S}\right)\mathrm{d}t. \quad (19)$$

For calculating the fluxes present in Eq. (19), we must determine which mesh we will perform the calculations on, whether the mesh Ω^n, or on Ω^{n+1}, or on several meshes at intermediate times between Ω^n and Ω^{n+1}.

If the mesh displacement is one-dimensional, as for smooth free-surface applications where the movement of the mesh can be limited to the same direction as gravity (typically vertical), without overturning, $\underline{v} = w\underline{e}_z$, the mesh velocity flux through cell faces can be discretized exactly as:

$$\begin{aligned}\Omega_c^{n+1} - \Omega_c^n &= \Delta t \sum_{f\in\mathcal{F}_c} \underline{v}_f\Big|_n^{n+1}\cdot\underline{S}_f^n \\ &= \sum_{f\in\mathcal{F}_c}\left(\underline{x}_f^{n+1} - \underline{x}_f^n\right)\cdot\underline{S}_f^n,\end{aligned} \quad (20)$$

where $\underline{x}_f^n$ is the face center of gravity at time t^n. This choice will be motivated and justified in Section 3.3.

Time integration of the conservation of mass and momentum, Eq. (16) and (18), gives:

$$M_c^{n+1} - M_c^n + \sum_{f\in\mathcal{F}_c}\int_{t^n}^{t^{n+1}} \left(\rho\left(\underline{u}-\underline{v}\right)\right)_f\cdot\underline{S}_f\mathrm{d}t = 0,$$

$$\underline{Q}_c^{n+1} - \underline{Q}_c^n + \sum_{f\in\mathcal{F}_c} \int_{t^n}^{t^{n+1}} \underline{u}_f \left(\rho\left(\underline{u}-\underline{v}\right)\right)_f \cdot \underline{S}_f \mathrm{d}t = \sum_{f\in\mathcal{F}_c} \int_{t^n}^{t^{n+1}} \underline{\underline{\sigma}}_f \cdot \underline{S}_f \mathrm{d}t. \quad (21)$$

It is important to remark here that ρ_c^{n+1} and $\underline{u}_c^{n+1}$ correspond to the mean density and the mean velocity of cell c at its location at time t^{n+1}, and no reconstruction is needed.

At this point one must choose a discretization in time to resolve the velocity field. As for the discrete volume conservation, Eq. (20), fluxes are written on the faces at time t^n.

The prediction step for the velocity, which was written for fixed meshes in Eq. (9), can now be written on moving meshes as:

$$M_c^n \frac{\widetilde{\underline{u}}_c^k - \underline{u}_c^n}{\Delta t} - \sum_{f\in\mathcal{F}_c} \widetilde{\underline{\underline{\tau}}}_f^k \cdot \underline{S}_f^n + \sum_{f\in\mathcal{F}_c} \left(\widetilde{\underline{u}}_f^k - \widetilde{\underline{u}}_c^k\right) \left(\rho\left(\underline{u}-\underline{v}\right)\right)_f \Big|_n^{n+1,k-1} \cdot \underline{S}_f^n = -\sum_{f\in\mathcal{F}_c} \left(p|_n^{n+1,k-1}\right)_f \underline{S}_f^n. \quad (22)$$

We see that pressure and friction forces are applied on the mesh at time n and mass flux takes face displacement into account. Again, the conservation of mass was used to transform the unsteady term and rearranged with the convective terms.

The correction step of the velocity, Eq. (11), remains unchanged on mesh at time t^n. At the end of this process, we obtain $\underline{u}_c^{n+1,k}$, $p_c|_n^{n+1,k}$, and the fluid mass flux $\left(\rho\underline{u}\right)_f \Big|_n^{n+1,k} \cdot \underline{S}_f^n$. Then, the mesh velocity is computed and the mass flux due to mesh displacement, $\left(\rho\underline{v}\right)_f \Big|_n^{n+1,k} \cdot \underline{S}_f^n$, is subtracted from it. The total mass flux is used in any transport equation afterwards. We see here that the mesh velocity has to be defined at the nodes of the mesh to move them, and it is also used when calculating the flux $\left(\rho\underline{v}\right)_f \Big|_n^{n+1,k-1} \cdot \underline{S}_f^n$ at cell faces in Eq. (22). Face mesh-velocity is deduced from mesh-velocities at the vertices by:

$$\underline{v}_f \Big|_n^{n+1} = \sum_{v\in\mathcal{V}_f} \frac{\underline{v}_v \Big|_n^{n+1} \cdot \underline{S}_{f,v}}{S_f^2} \underline{S}_f, \quad (23)$$

where $\underline{S}_{f,v}$ is the portion of face f vector associated to vertex v displayed in Fig. 1.

3.3. Verification of the discretized geometric conservation law

The geometric conservation law (GCL) states that a variation of volume Ω_c of a cell c between time-step t^n and t^{n+1} should be equal to the integral over this time interval of the mesh velocity flux through faces of cell c:

$$\Omega_c^{n+1} - \Omega_c^n = \int_{t^n}^{t^{n+1}} \int_{\partial\Omega_c} \underline{v} \cdot \mathrm{d}\underline{S}\, \mathrm{d}t. \quad (24)$$

The time integration scheme used for the mesh velocity allows one to rewrite the time integral as:

$$\int_{t^n}^{t^{n+1}} \int_{\partial\Omega_c} \underline{v} \cdot \mathrm{d}\underline{S}\, \mathrm{d}t = \Delta t \int_{\partial\Omega_c^n} \underline{v}|_n^{n+1} \cdot \mathrm{d}\underline{S}^n. \quad (25)$$

It is clear that the condition in Eq. (24) does not hold true if the mesh velocity $\underline{v}$ contains more than one direction of variation. The integration scheme must be modified if we want to satisfy the GCL for mesh velocities which vary in two or three directions [see 21, for a presentation of these types of schemes and their properties].

For one-dimensional mesh movement in z, one sees that $\left(\underline{x}_f^{n+1} - \underline{x}_f^n\right) \cdot \underline{S}_f^n = \left(z_f^{n+1} - z_f^n\right) S_{f,z}^n$, where $S_{f,z}$ is the z component of $\underline{S}$. Hence, volume of cell c can be computed

as:

$$\Omega_c = \int_{\Omega_c(t)} \text{div}\ (0,\, 0,\, z) \tag{26}$$

$$= \sum_{f\in\mathcal{F}_c} z_f S_{f,z}, \tag{27}$$

and one can prove that $S_{f,z}$ is constant over time. Indeed, we can first remark that $\underline{S}_f$ can be defined using only surrounding vertices of f, even if the face is warped. Formulae proposed by [10] for a hexahedron can be extended for any polyhedron. We can remark that $S_{f,x}$ can be computed as $S_{f,x} = \int_f \underline{curl}\, \underline{r}_x \cdot \underline{\mathrm{d}S} = \oint_{\partial f} \underline{r}_x \cdot \underline{\mathrm{d}l}$ with $\underline{r}_x = \frac{1}{2}(0,\, -z,\, y)$ such that $\underline{curl}\, \underline{r}_x = (1,\, 0,\, 0)$. Similarly, $S_{f,y} = \int_f \underline{curl}\, \underline{r}_y \cdot \underline{\mathrm{d}S} = \oint_{\partial f} \underline{r}_y \cdot \underline{\mathrm{d}l}$ and $S_{f,z} = \int_f \underline{curl}\, \underline{r}_z \cdot \underline{\mathrm{d}S} = \oint_{\partial f} \underline{r}_z \cdot \underline{\mathrm{d}l}$ with $\underline{r}_y = \frac{1}{2}(z,\, 0,\, -x)$ and $\underline{r}_z = \frac{1}{2}(-y,\, x,\, 0)$ such that $\underline{curl}\, \underline{r}_y = (0,\, 1,\, 0)$ and $\underline{curl}\, \underline{r}_z = (0,\, 0,\, 1)$. So $S_{f,z}$ depends only on the x and y components of the vertices of the surrounding edges of face f. Consequently, Eq. (20) is exactly fulfilled if faces remain planar. If the mesh displacement is 2-D or 3-D, the mesh velocity fluxes must be split into two or three steps to exactly conserve the volume [21]. Note also that the calculation of $\underline{S}_f$ as a curl ensures that for any cell c, $\sum_{f\in\mathcal{F}_c} \underline{S}_f = \underline{0}$ even for warped faces.

3.3.1. Conservation of a uniform velocity field

The time integration scheme chosen has to have good properties to maintain a uniform velocity field. This point is key if we would like to have a stable scheme and verify the maximum principle [see 21].

Let us consider the numerical scheme of the prediction step, Eq. (22), rewritten in velocity increment $\delta\underline{u} = \widetilde{\underline{u}}^k - \underline{u}^n$ for a constant pressure field:

$$\frac{\boldsymbol{M}_c^n}{\Delta t}\delta\underline{u}_c + \sum_{f\in\mathcal{F}_c}\left(\delta\underline{u}_f - \delta\underline{u}_c\right)\left(\rho\left(\underline{u}-\underline{v}\right)\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f^n = -\sum_{f\in\mathcal{F}_c}\left(\underline{u}_f^n - \underline{u}_c^n\right)\left(\rho\left(\underline{u}-\underline{v}\right)\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f^n. \tag{28}$$

We have not considered diffusive terms, but this does not change the conclusions here. From this, it is easy to verify that for a uniform field $\underline{u}^n$, the right-hand-side of Eq. (28) is zero. For an upwind convective scheme on the velocity increment, the system to solve is written as:

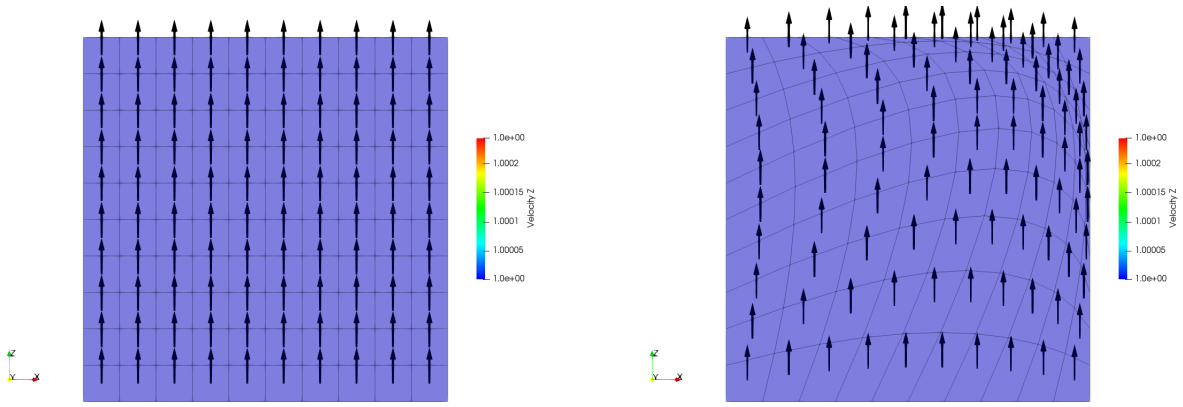
$$\underline{\underline{EM}}\,\delta\underline{u}^{n+1} = \underline{0}, \tag{29}$$

with $EM_{cc} = \frac{M_c^n}{\Delta t} + \sum_{f\in\mathcal{F}_c}\max\left(-\left(\rho\left(\underline{u}-\underline{v}\right)\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f^n,\, 0\right)$, and for $c\neq j$ and f the face between c and j, $EM_{cj} = -\max\left(\left(\rho\left(\underline{u}-\underline{v}\right)\right)_f\Big|_n^{n+1,k-1}\cdot\underline{S}_f^n,\, 0\right)$. One can note that $\underline{\underline{EM}}$ is a M-matrix (for all $c\neq j$, $EM_{cj}\leq 0$ and for all c, $EM_{cc} > \sum_j \left|EM_{cj}\right|$) and is therefore invertible.

Just like with a uniform zero velocity, the numerical verification of this assertion is presented in Fig. 2, where an initial field $\underline{u} = \underline{e}_z 1.0\,\mathrm{m/s}$ is not modified by the mesh displacement imposed for all nodes by:

$$\frac{\underline{x}-\underline{x}_0}{L} = 4\sin\left(\frac{2\pi t}{T}\right)\frac{x_0}{L}\left(1-\frac{x_0}{L}\right)\frac{z_0}{L}\left(1-\frac{z_0}{L}\right)\begin{pmatrix}1\\0\\1\end{pmatrix}, \tag{30}$$

where $L = 1\,\mathrm{m}$ is also the size of the square computation domain, and $T = 4\,\mathrm{s}$ is the period of the mesh velocity variations.



(a) Mesh and velocity field at initial time.

(b) Mesh and velocity field after 1 s.

Figure 2: Conservation of a uniform velocity field despite a two-dimensional deformation of the mesh given by Eq. (30).

3.4. Vector solving of the mesh velocity

Up to this point, we have considered a mesh velocity only to verify the boundary conditions of the moving mesh, but any arbitrary velocity that satisfies these conditions is acceptable. Here, the mesh velocity is given by solving a Poisson equation on $\Omega(t)$:

$$\underline{\underline{\mathrm{div}}}\left(\underline{\underline{\lambda}}\,\underline{\underline{\nabla}}\,\underline{v}\right) = \underline{0}, \tag{31}$$

$$\underline{v}\big|_{\partial\Omega^{imp}(t)} = \text{Imposed velocity}, \tag{32}$$

$$\left.\frac{\partial\left(\underline{v} - \underline{v}\cdot\underline{n}\,\underline{n}\right)}{\partial n}\right|_{\partial\Omega\setminus\partial\Omega^{imp}(t)} = \underline{0}, \tag{33}$$

$$\underline{v}\cdot\underline{n}\big|_{\partial\Omega\setminus\partial\Omega^{imp}(t)} = 0, \tag{34}$$

where $\partial\Omega^{imp}(t)$ represents the boundary of the domain with either a free-surface or imposed velocity, and where $\partial\Omega \setminus \partial\Omega^{imp}(t)$ is the fixed boundary where we allow the mesh to slide along this boundary. Of course one could also impose the mesh displacement within the domain, or even zero (a fixed mesh), or impose a sliding condition on a free-surface (*i.e.*, to fix a non-zero value of $\underline{v}\cdot\underline{n}$ and fix a Neumann condition for the tangential part of the mesh velocity on the boundary).

The tensor $\underline{\underline{\lambda}}$ is normally taken to be the identity matrix, but an anisotropic heterogeneous tensor can be chosen to make the mesh more rigid in a particular direction and in a given region. Note that having an anisotropic $\underline{\underline{\lambda}}$ induces a strong coupling between the components of the mesh velocity $\underline{v}$ and therefore requires a coupled solver.

At this stage, it is important to note that it is more natural to solve Eq. (31) with each Degree of Freedom (DoF) stored at vertices of the mesh. *Code_Saturne* legacy schemes are cell-based with the mesh velocity $\underline{v}_c$ stored at the cell centers. Therefore, once $\underline{v}_c$ is calculated, nodes displacement must be deduced by interpolation.

Code_Saturne can make use of algorithms based on the Compatible Discrete Operators (CDO) framework [7], and DoFs can be either at cell centers (cell-based schemes) or at vertices (vertex-based schemes). A CDO vector vertex-based solver on the dual barycentric mesh is thus used to solve Eq. (31) and is compared to the legacy approach detailed hereafter. Note that, thanks to the mapping $\underline{x}_0 \rightarrow \underline{x}\left(\underline{x}_0, t\right)$, solving Eq. (31) on $\Omega(t)$ can also be solved on the original domain $\Omega(0)$. This is the choice made when using CDO schemes for the mesh velocity, to save time on the computation of additional mesh quantities required by these

schemes.

3.4.1. Free-surface boundary condition

The next step is to make use of the ALE module for modeling a free-surface. This work is similar to what is performed in other models like Telemac3D [28], in the sense that the computational domain evolves with the free-surface, but contrary to Telemac3D, fluid velocity and pressure are solved with a cell-based finite volume scheme and not with a vertex based finite element scheme, and domain is meshed using cells of any star-shaped polyhedral type.

The aim of the proposed algorithm is to solve mesh velocity after solving fluid quantities to impose correctly the kinematic condition at the free-surface without any sub-iterations. In addition, the mesh-velocity solver is component-coupled so that sliding conditions are taken into account in an exact manner.

To use the ALE module to simulate a free-surface, we first must specify the free-surface boundary conditions for both the fluid and mesh velocities. For the first one, in the absence of wind, a homogeneous Neumann condition is applied on the fluid as well as a constant pressure p_0:

$$\left.\frac{\partial \underline{u}}{\partial n}\right|_{\text{surface}} = \underline{0},\ p = p_0. \tag{35}$$

It would be possible to take into account wind by providing a shear stress between the air and water (see [28] for an example of this approach). For simulations with turbulence, it is also necessary to impose conditions for the turbulent quantities.

The aim is to impose a kinematic condition for the mesh velocity:

$$\underline{v} \cdot \underline{n}\big|_{\text{surface}} = \underline{u} \cdot \underline{n}\big|_{\text{surface}}. \tag{36}$$

For now we assume that the mesh velocity is unidirectional, for example $\underline{v} = w\underline{e}_z$. This assumption is done following the remarks in 3.3 concerning the GCL.

Since mass flux is known explicitly when solving the mesh velocity, the flux $\underline{u} \cdot \underline{n}$ is used as a boundary condition for the mesh velocity, as summarized below in Section 3.5.

3.4.2. Legacy approach: node displacements from the mesh velocity

In this section, cell-based mesh velocity $\underline{v}_c$ is obtained with the vector coupled-solver and will be denoted by the "legacy" version.

We differentiate first the nodes on the external boundaries (called the exterior nodes) and others (called interior nodes). We specify here that the nodes which are on a sliding surface are considered internal, with a specific treatment detailed hereafter. Thus, all nodes belonging to $\partial\Omega^{imp}(t)$ are considered as external, all the others as internal.

Treatment of internal nodes For an interior node v, we note $\mathcal{C}_v$ as the list of cells c which include node v. If $\underline{x}_c$ is designated the center of a cell $c \in \mathcal{C}_v$, the velocity $\underline{v}_v$ of v is defined by:

$$\underline{v}_v = \frac{\displaystyle\sum_{c\in\mathcal{C}_v} \frac{1}{\Omega_c}\left(\underline{v}_c + \underline{\underline{\nabla}}\,\underline{v}_c \cdot \left(\underline{x}_v - \underline{x}_c\right)\right)}{\displaystyle\sum_{c\in\mathcal{C}_v} \frac{1}{\Omega_c}}. \tag{37}$$

Treatment of exterior nodes For an exterior node v, we set $\mathcal{F}_v$ as the list of external faces where v appears. If $\underline{x}_f$ is designated the center of a face $f \in \mathcal{F}_v$ and $\underline{x}_c$ is the center

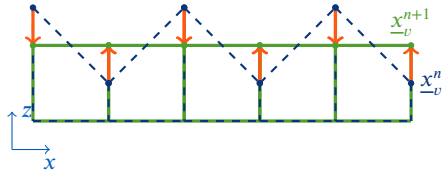


Figure 3: Illustration of the effect of the filtering (43) on the standing wave test case described in Section 4.2 with an initial wavelength of $2\Delta x$ planned in one time step. Without filtering, no fluid or mesh motion is obtained.

of the cell where f appears, we then have a velocity $\underline{v}_v$ of v defined by:

$$\underline{v}_v = \frac{\displaystyle\sum_{f\in\mathcal{F}_v} \frac{1}{|S_f|}\left(\underline{v}_f + \underline{\underline{\nabla}}\,\underline{v}_{\underline{x}_c}\cdot\left(\underline{x}_v - \underline{x}_f\right)\right)}{\displaystyle\sum_{f\in\mathcal{F}_v}\frac{1}{|S_f|}}. \tag{38}$$

Treatment of internal nodes at a sliding boundary For an internal node v already treated by Eq. (37) which also appears on a sliding face f and with an exterior normal vector $\underline{n}$, we project $\underline{v}_v$ on the face f as follows:

$$\underline{v}_v \leftarrow \left(\underline{\underline{1}} - \underline{n}\otimes\underline{n}\right)\underline{v}_v = \underline{v}_v - \left(\underline{v}_v\cdot\underline{n}\right)\underline{n}. \tag{39}$$

3.4.3. CDO approach: node displacements from the mesh velocity

In this section, mesh DoFs of the mesh velocity are directly associated to the nodes and the main advantage is that no interpolation is needed to displace nodes:

$$\underline{x}_v^{n+1} = \underline{x}_v^n + \Delta t\ \underline{v}_v\Big|_n^{n+1}. \tag{40}$$

At the same time, boundary conditions on primal boundary faces have to be translated in terms of node mesh-velocity $\underline{v}_v$ on the dual mesh for the free-surface condition. For a node v at the free-surface, we set $\mathcal{F}_v$ as the list of external free-surface faces where v appears, and for each face f of $\mathcal{F}_v$, c_f is the boundary cell to which f belongs to. The node mesh-velocity is therefore defined as:

$$\underline{v}_v\Big|_n^{n+1} = \frac{\displaystyle\sum_{f\in\mathcal{F}_v}\frac{\rho\underline{u}_f\Big|_n^{n+1}\cdot\underline{n}}{\rho_{c_f}}\underline{S}_{f,v}\cdot\underline{e}_z}{\displaystyle\sum_{f\in\mathcal{F}_v}\underline{S}_{f,v}\cdot\underline{e}_z}\underline{e}_z. \tag{41}$$

Eq. (41) does not ensure local conservation, because the face center mesh velocity $\underline{v}_f\Big|_n^{n+1}$, deduced from the nodes mesh velocity by (23) will not give back $\dfrac{\rho\underline{u}_f\Big|_n^{n+1}\cdot\underline{n}}{\rho_{c_f}}$. However, global conservation is ensured:

$$\sum_{f\in\text{free-surface}}\rho_{c_f}\,\underline{v}_f\Big|_n^{n+1}\cdot\underline{S}_f = \sum_{f\in\text{free-surface}}\rho\underline{u}_f\Big|_n^{n+1}\cdot\underline{S}_f. \tag{42}$$

Repartition of mass fluxes from faces to nodes written in Eq. (41) can be modified without breaking the global conservation (42). Indeed, if we add any affine function of zero mean over face f, (42) still holds. This property is convenient to filter modes such as waves of length $\lambda = 2\Delta x$ where Δx is the horizontal space grid, or to smooth too steep waves before they break (from numerical, rather than physical, reasons). We modify Eq. (41) by:

$$\underline{v}_v\Big|_n^{n+1} = \frac{\displaystyle\sum_{f\in\mathcal{F}_v}\left(\frac{\rho\underline{u}_f\Big|_n^{n+1}\cdot\underline{n}}{\rho_{c_f}} - \alpha_f\frac{z_v - z_f}{\Delta t}\right)\underline{S}_{f,v}\cdot\underline{e}_z}{\displaystyle\sum_{f\in\mathcal{F}_v}\underline{S}_{f,v}\cdot\underline{e}_z}\underline{e}_z, \tag{43}$$

where α_f is a factor equal to 0 or 1 to activate the filter only when necessary. α_f is set to 1 for face f when there is at least one of its nodes which is a local maximum (higher than all

the neighboring face centers) and one of its nodes which is a local minimum (this criterion correspond to the characterization of a wavelength $2\Delta x$). α_f is also set to 1 for face f which is too steep (if one edge v_0v_1 is such that Δx is smaller than Δz, where Δx and Δz are defined by:

$$\begin{aligned} \Delta x &= \left\| \left(\underline{\underline{1}} - \underline{e}_z \otimes \underline{e}_z \right) \cdot \left(\frac{\underline{x}_{v_0} + \underline{x}_{v_1}}{2} - \underline{x}_f \right) \right\|, \\ \Delta z &= \left| \left(\frac{\underline{x}_{v_0} + \underline{x}_{v_1}}{2} - \underline{x}_f \right) \cdot \underline{e}_z \right|. \end{aligned} \tag{44}$$

Term $\alpha_f \frac{z_v - z_f}{\Delta t}$ acts as a volume-conservative smoother for nodes that have an elevation different from the weighted mean elevation of the surrounding faces. The behavior of the filtering given by Eq. (43) is illustrated in Fig. 3 where the free-surface is initialized with a sawtooth elevation of $2\Delta x$ wavelength, but quiescent domain (with zero fluid velocity). In this case, water is supposed to be at rest and elevation modes cannot be suppressed by the legacy scheme. The filtering is removing it in one step, without changing the potential energy based on the elevation of free-surface faces. Similar types of instabilities have been known since Longuet-Higgins and Cokelet [36]. More sophisticated approaches to removing this issue have been illustrated by [41].

3.5. Summary of the ALE algorithm

The algorithm described above to solve the Navier-Stokes equations on a moving mesh reads:

for time interval n to $n+1$ **do**

  for sub-iteration k **do**

    Solve the predicted velocity $\widetilde{\underline{u}}_c^{n+1,k}$ on Ω^n;

    Calculate corrected velocity $\underline{u}_c^{n+1,k}$ on Ω^n and the flux $\underline{u}_f\Big|_n^{n+1,k} \cdot \underline{S}^n$;

    if using legacy solver for $\underline{v}$ **then**

      Solve $\underline{v}_c\Big|_n^{n+1,k}$ on the primal mesh Ω^n with boundary conditions (36);

      Reconstruct velocity from cell centered values at nodes $\underline{v}_v\Big|_n^{n+1,k}$;

    else if using CDO solver for $\underline{v}$ **then**

      Calculate $\underline{v}$ boundary conditions at nodes;

      Solve $\underline{v}_v\Big|_n^{n+1,k}$ on the original dual mesh Ω^0 with boundary conditions (36);

    end if

    Update mesh from Ω^n to $\Omega^{n+1,k}$ using $\underline{v}_v\Big|_n^{n+1,k}$;

    Calculate total mass flux $\rho\left(\underline{u} - \underline{v}\right)_f\Big|_n^{n+1,k} \cdot \underline{S}^n$;

  end for

end for

4. Verification and Validation Test Cases

In order to validate these developments for modeling wave propagation and wave-structure interaction, we consider a variety of verification and validation test cases, showing the fluid motion generated by a piston in order to validate the ALE module without free-surface condition; standing waves in a closed basin with small and high initial wave amplitudes to verify volume conservation; the viscous damping of waves, in order to validate the proper rate of decay due to viscosity; the propagation of a solitary wave, showing the exact reproduction of a large amplitude wave; the propagation of waves over a submerged bar, showing the ability to handle shoaling; the force on a vertical cylinder in regular waves, showing issues of wave-structure interaction; and finally comparisons against a novel type of wave-maker.

4.1. Test case of a 2-D axisymmetric piston

This case consists in studying a fluid flow inside a cylindrical valve (piston) with a sudden expansion. Numerical results are compared to experimental data from [18]. The simulated domain is axisymmetric. The flow is laminar ($Re \simeq 320$), with a density of $\rho = 1000\,\mathrm{kg/m^3}$ and a dynamic viscosity $\mu = 4 \times 10^{-3}\,\mathrm{kg/m/s}$.

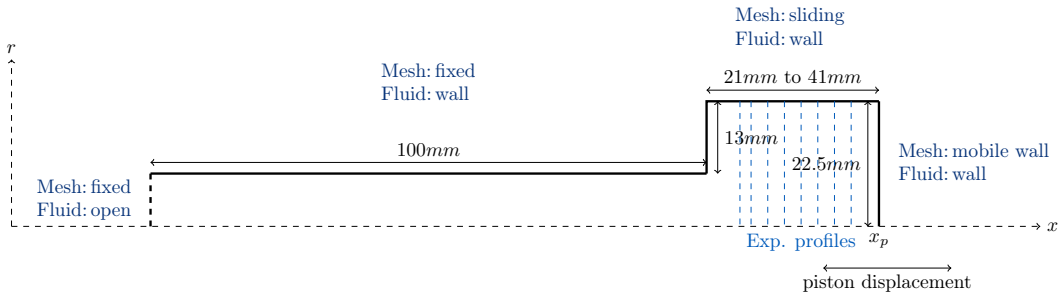


Figure 4: Sketch of the valve validation case.

Time	Piston velocity	Piston displacement
$0 \le t \le T_i$	$v_p = \frac{V_i}{T_i} t$	$d_p = \frac{V_i}{T_i} \frac{t^2}{2}$
$T_i \le t \le T_{TDC} - T_i$	$v_p = V_i$	$d_p = X_i + V_i \left(t - T_i\right)$
$T_{TDC} - T_i \le t \le T_{TDC}$	$v_p = \frac{V_i}{T_i} \left(T_{TDC} - t\right)$	$d_p = 2X_i + V_i \left(T_{TDC} - 2T_i\right)$ $-\frac{V_i}{T_i} \frac{\left(T_{TDC} - t\right)^2}{2}$
$T_{TDC} \le t \le T_{TDC} + T_c$	$v_p = -\frac{V_c}{T_c} \left(t - T_{TDC}\right)$	$d_p = 2X_i + V_i \left(T_{TDC} - 2T_i\right)$ $-\frac{V_c}{T_c} \frac{\left(t - T_{TDC}\right)^2}{2}$
$T_{TDC} + T_c \le t \le T_{BDC} - T_c$	$v_p = -V_c$	$d_p = 2X_i + V_i \left(T_{TDC} - 2T_i\right)$ $-x_2 - V_c \left(t - T_{TDC} - T_c\right)$
$T_{TDC} - T_c \le t \le T_{BDC}$	$v_p = -\frac{V_c}{T_c} \left(T_{BDC} - t\right)$	$d_p = \frac{V_c}{T_c} \frac{\left(t - T_{TDC}\right)^2}{2}$

Table 1
Piston prescribed velocity and displacement.

Fig. 4 describes the geometry and boundary conditions for the Navier-Stokes equations and the mesh velocity. Two different kinds of "mobile wall condition" are implemented in *Code_Saturne*:

- imposed velocity of the interface at the center of each boundary face;
- given displacement of each node of the mobile interface.

When imposing piston motion, it is decomposed into six steps in which either the piston velocity v_p or the piston displacement d_p are prescribed, as summarized in Table 1.

The piston extension is maximum at time T_{TDC} (Top Dead Center, $x_p = X_{max} = 0.141$ m) and minimum at time T_{BDC} (Bottom Dead Center, $x_p = X_{min} = 0.121$ m) given in Eq. (45). The maximum velocity during the injection phase is $V_i = 0.006\,22$ m/s, and the time during the acceleration phase in which the velocity passes from zero to V_i is given by $T_i = 0.05$ s. The minimum velocity during the compression phase is $V_c = 0.0265$ m/s, and the time during the acceleration phase in which the velocity passes from zero to $-V_c$ is given by $T_c = 0.18$ s where $X_i = \frac{V_i \times T_i}{2}$, $X_c = \frac{V_c \times T_c}{2}$. Note that for the maximum expansion time and the complete compression time the following equations holds:

$$\begin{aligned} T_{TDC} &= \frac{X_{max} - X_{min} + 2X_i}{V_i} \simeq 3.265\,\text{s}, \\ T_{BDC} &= T_{TDC} + \frac{X_{max} - X_{min} + 2X_c}{V_c} \simeq 4.200\,\text{s}. \end{aligned} \tag{45}$$

Selected axial velocity profiles are respectively displayed in Fig. 6 for the compression step, in comparison to experimental data, showing good agreement. First, one can notice that same results are obtained with the CDO and legacy

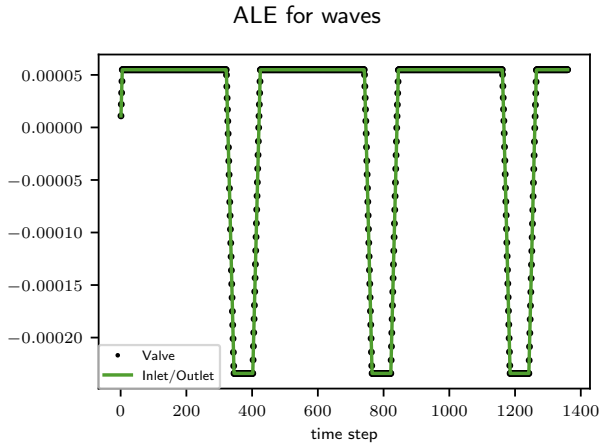


Figure 5: Comparison of mass flux computed at the open boundary (green) and the value imposed by the piston, represented in black dots.

solver for the mesh velocity, and from imposing mesh displacement or directly the mesh velocity. This is expected because the code transforms the mesh displacement boundary condition into a mesh velocity boundary condition to compute it. One can also verify that the incoming and outgoing mass flux should be equal to the imposed mass flux due to the valve displacement. As the flow is incompressible, the two fluxes displayed in Fig. 5 must be equal.

4.2. Standing wave in a closed basin

This case is initialized by a free-surface with a sinusoidal form with a wavelength λ. We consider a basin with length $L = 400\,\mathrm{m}$, and average depth $h_0 = 10\,\mathrm{m}$ and we fix $\lambda = \frac{L}{2}$. The domain is two-dimensional (*i.e*, one cell thick) and the mesh used is composed of 10 cells in the z direction and 400 in the x direction.

In order to fix a period of the wave close to 10 s, we take the acceleration due to gravity to be $g = 40\,\mathrm{m/s^2}$.

(Alternatively, we could start from a nonlinear standing wave solution instead of a pure sinusoidal initial condition, but our goal here is not to look at the nonlinear wave dynamics, which we will consider in the next cases, but simply to evaluate the volume conservation and stability.)

The free-surface deforms over time, and with an initial amplitude A, one or several (particularly for large amplitude initial conditions) modes become visible. To evaluate the numerical algorithm used, we begin by studying a wave which is quasi-linear.

Linear standing wave When $\frac{A}{h_0} \ll 1$, linear wave theory can be used to describe the free-surface, $z_{surf}\,(x,\,t)$, which is given by:

$$z_{surf}\,(x,\,t) = A\cos\left(\frac{2\pi x}{\lambda}\right)\cos\left(\frac{2\pi t}{T}\right), \tag{46}$$

where the period of the wave is given by the celerity c, as $T = \frac{\lambda}{c}$, with the dispersion relation giving:

$$c_0^2 = g\frac{\lambda}{2\pi}\tanh\left(\frac{2\pi h_0}{\lambda}\right). \tag{47}$$

In fact we consider a shallow water case, where $h \ll \lambda$, so we can approximate this as $c_0 \simeq \sqrt{g h_0}$. We obtain precisely:

$$c_0 = \sqrt{g\frac{\lambda}{2\pi}\tanh\left(\frac{2\pi h_0}{\lambda}\right)} \simeq 19.681\,\mathrm{m/s}, \tag{48}$$

giving $T_0 = \frac{\lambda}{c_0} \simeq 10.162\,\mathrm{s}$.

We choose to start with a case with wave amplitude $A = 0.025\,\mathrm{m}$. The simulation of one hundred periods is shown in Figure 7 where we can observe the evolution of the wave height in time at the point $x = 0$ (the center of the domain).

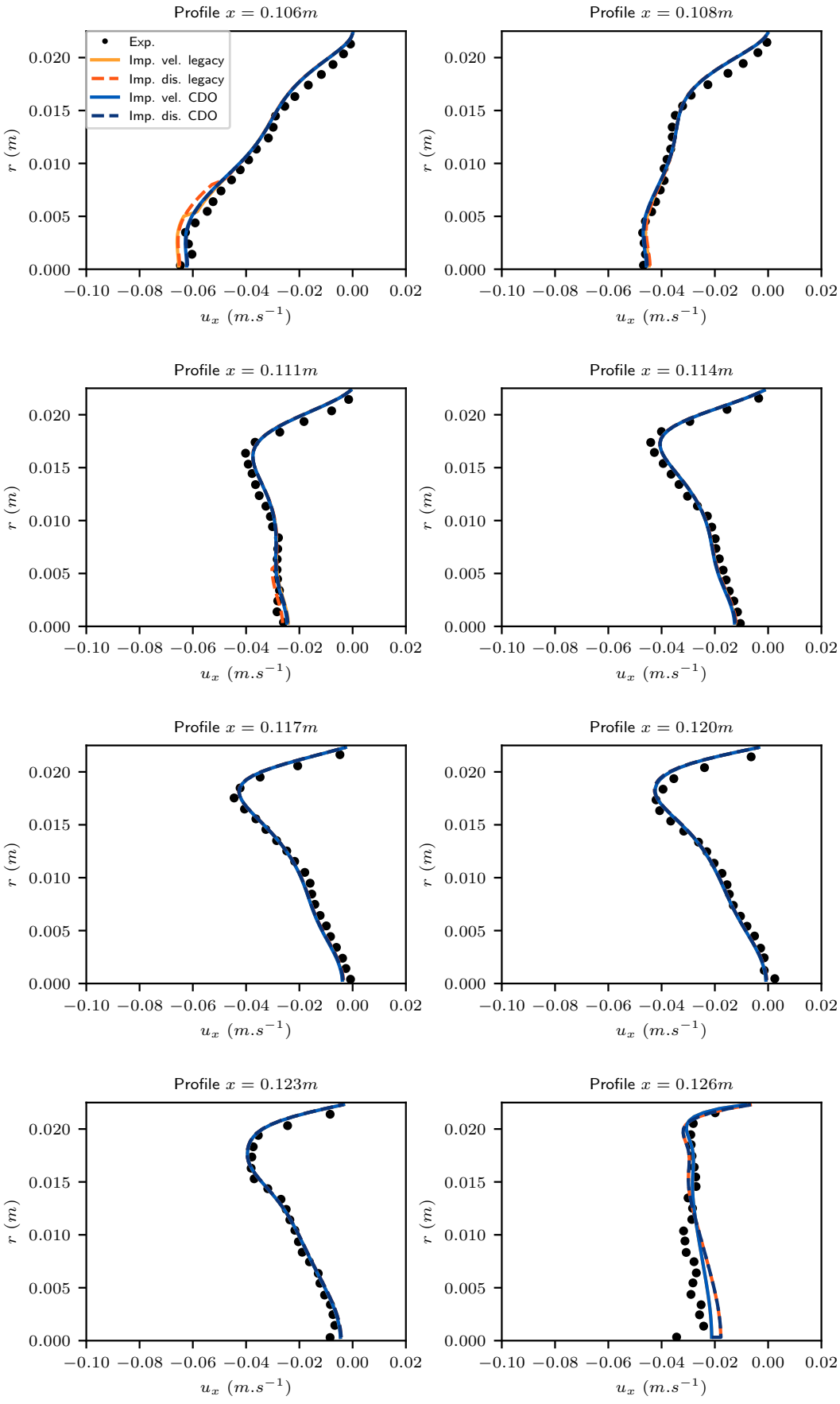
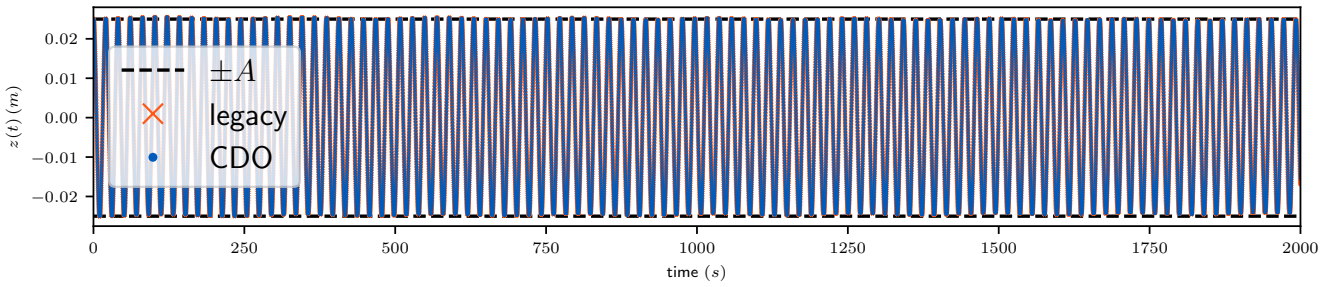
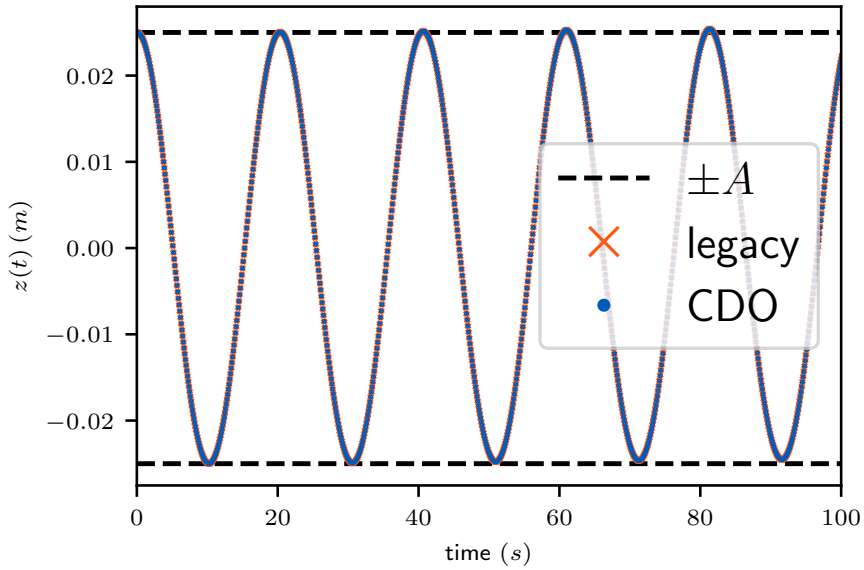


Figure 6: Axial velocity profiles at different positions x for the compression step ($x_p = 0.127\,\mathrm{m}$).



(a) 100 periods.



(b) Zoom over 5 periods.

Figure 7: Time series of wave elevation at the center point in the case of a standing wave. The initial wave amplitude is given as $A = 0.025\,\mathrm{m}$ for an average water depth of $h_0 = 10\,\mathrm{m}$.

The period obtained numerically is $T_{num} = 10.163 \pm 0.001$ s (the range obtained is taken to be the time-step divided by the number of periods observed). This is therefore a good match for the theoretical T_0, which we see in Fig. 7, $T = T_0 \pm 0.025$ s.

We note that in Fig. 7, that for a relatively long duration, the wave height remains nearly constant which indicates that the model is not losing significant amounts of volume or energy.

Nonlinear standing waves and the conservation of volume When the amplitude A of the initial sinusoidal deformation is increased, linear theory is no longer valid, and other modes of oscillation can appear. The goal here is not to study the other modes, but to determine the validity of the numerical developments presented here. As the initial amplitude increases from $A = 0.025\,\mathrm{m}$ up to $A = 2.5\,\mathrm{m}$ (a case that was numerically unstable in earlier versions of *Code_Saturne*), we can consider the relative variation in volume over time.

We recall that from §3.3, this volume numerically does not necessarily remain constant using the legacy approach because of the interpolations between nodes and cell centers cause differences in how the mesh and fluid velocity are considered. On the contrary, the CDO approach shows a good total volume conservation over time. The error remains under the precision used to solve the mass equation.

In Figure 9, we see that the relative variation in volume

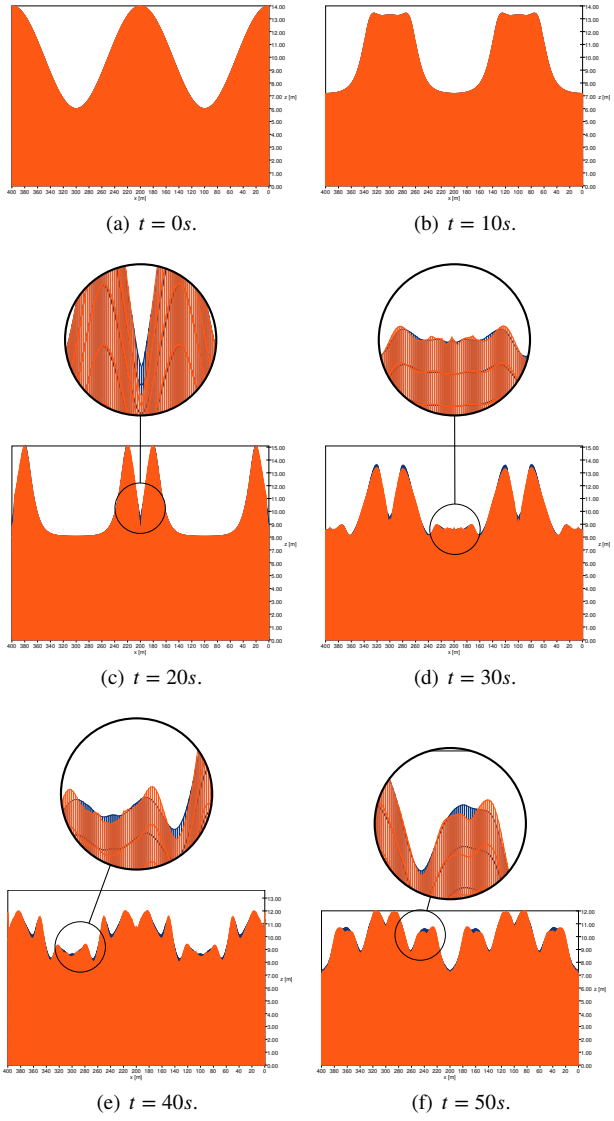
for all amplitudes with the CDO approach does not exceed 10^{-9}, which is smaller mass solver precision (10^{-8} is the tolerance used here for the pressure solver). As expected, for larger wave amplitudes with the legacy approach, the loss of volume becomes larger, but this variation in volume seems to stabilize after an initial adjustment period. For example, for an amplitude of 2.5 m, the quasi-total loss of volume is made at the beginning of the simulation, when the wave of 2.5 m breaks into a number of other wavelengths.

The most non-linear case can also be used to see the effect of the filtering introduced with the CDO solver (see Figure 8).

4.3. Viscous damping of waves

To verify that the application of these ALE schemes has not affected the ability of the algorithm to properly model viscous effects, we next consider a standing wave, similar to the previous section, but with viscosity added. Though this can be extended to arbitrary depths, as in Antuono and Colagrossi [1], we consider a wave which normally would be considered a deep water case, and then can use the solution from [34] to describe the viscous decay.

The domain we consider is again two-dimensional, 128×64 cells (in the x- and z-directions, respectively, with a width of $L = 2\,\mathrm{m}$ and depth $h_0 = 1\,\mathrm{m}$. The bottom boundary-condition is no-slip, but the side boundaries are free-slip. For this test case, we take $\rho = 1\,\mathrm{kg\,m^3}$ and $g = 1\,\mathrm{m\,s^{-2}}$, to make all variables non-dimensional. A time-step of $\sqrt{\pi}/50\,\mathrm{s}$ was used, which corresponds to $\frac{1}{100}^{th}$ of the period of oscillation for an inviscid linear wave. Instead of having an initial free-surface elevation, here we impose on an undisturbed domain the velocity field of a standing Airy wave solution:



(a) $t = 0s$. (b) $t = 10s$. (c) $t = 20s$. (d) $t = 30s$. (e) $t = 40s$. (f) $t = 50s$.

Figure 8: Snapshots of free-surface for the filtered (in blue) and the non-filtered (in orange) for the standing wave test case with an initial $A = 4\,\mathrm{m}$ and an average depth of $h_0 = 10\,\mathrm{m}$; a close-up with three times magnification is added to emphasise the differences, mainly next to the pics.

$$\underline{u}\left(\underline{x}, t=0\right) = \begin{pmatrix} \omega A \sin kx \dfrac{\cosh\left(k(z+h_0)\right)}{\sinh\left(kh_0\right)} \\ 0 \\ -\omega A \cos kx \dfrac{\sinh\left(k(z+h_0)\right)}{\sinh\left(kh_0\right)} \end{pmatrix}. \quad (49)$$

Taking $kh_0 = \pi$, and a wave amplitude of $A = 0.005h_0$, the dispersion relationship, $\omega^2 = gk \tanh kh_0$, reduces to

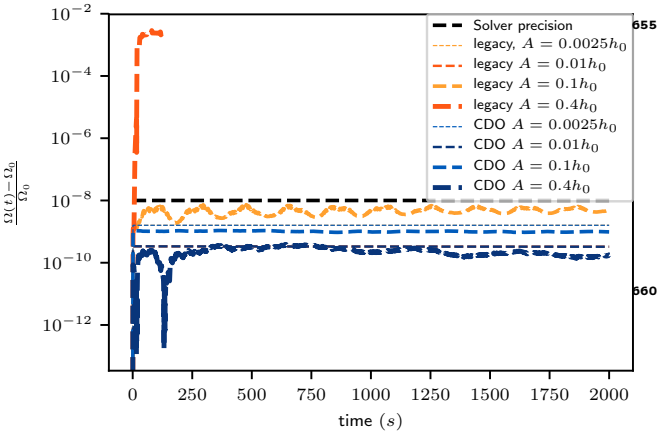


Figure 9: Relative change in volume in terms of time for the standing wave test case with different initial wave heights, ranging from $A = 0.0025h_0$ to $A = 0.4h_0$ with an average depth of $h_0 = 10\,\mathrm{m}$. Dashed line shows the precision of the solver for pressure (10^{-8}).

the deep-water case. After, we examine the evolution of the kinetic energy, $\mathcal{E}_K(t) = \int_\Omega ||\underline{u}||^2/2\mathrm{d}\Omega$, compared to the expected exponential decay, where $\mathcal{E}_K(t) \propto e^{-\beta\sqrt{\frac{g}{h_0}}t}\mathcal{E}_K(0)$ for a coefficient β.

In deep water, the viscous dissipation of waves due to the viscosity of the fluid has been studied since the work of [34], who predicted $\beta = 4\pi^2/Re$. The exact rate of decay is slightly different in this case, because for low Reynolds numbers (where $Re = h_0\sqrt{gh_0}/\nu$), this classic solution does not take into account the bottom boundary layer, and even though for inviscid waves, for wave-numbers $kh_0 \geq \pi$, we are in the deep-water regime, a more complete solution was provided by Antuono and Colagrossi [1], who solved the linearized Navier-Stokes solution that included the effect of the bottom boundary layer, thus $\beta = 4\pi^2 Re^{-1} - 2\sqrt{2}\pi^{11/4}Re^{-1.5}$, and showed results that more closely match what is obtained here (Fig. 10). In this case, little difference is seen between legacy and CDO solvers, as the amplitude of motion is very small.

4.4. Solitary wave

In addition to handling a moving free-surface, for ocean engineering applications it is important to be able to generate waves, particularly steep waves. One form of this which can easily be studied mathematically is a solitary wave, where the wave translates over constant depth without changing shape.

Here we consider a domain $L = 40\,\mathrm{m}$ long, a depth $h_0 = 1\,\mathrm{m}$, and a wave with moderately large relative height $A/h_0 = 0.6$. Viscosity is reduced to a negligible value, so any changes to the wave shape in the domain should be due entirely to numerical errors. Again we consider a non-dimensional case, with $\rho = 1\,\mathrm{kg\,m^{-3}}$, $g = 1\,\mathrm{m\,s^{-2}}$, $h_0 = 1\,\mathrm{m}$.

On the wave-maker (left) boundary, the solution of the Euler equations is computed using the highly accurate solution of Dutykh and Clamond [19] to impose the horizontal and vertical velocity fields as a boundary condition. In order to start with a wave-tank initially at rest, the wave is shifted in time so that the wave crest passes $x = 0\,\mathrm{m}$ at $t = 8.0\,\mathrm{s}$.

On the other (right) side, an Orlanski-type radiative outlet is imposed, where all variables (*i.e.*, velocity and pressure) are assumed to propagate at the theoretical celerity of the wave; see Orlanski [42] or the *Code_Saturne* theory documentation [46] for more detail. Note that this outlet condition is only useful for this case, where the solitary wave-speed is known. For more general cases where different waves propagate at different speeds, other damping methods are required, as discussed in the next application.

We are able to reproduce a solitary wave elevation within a few percent at a gauge in the middle of our domain (Fig. 11). The reflection is of similar magnitude, and likely due partly to an inaccurate wave celerity (*i.e.*, the speed of the numerically modeled wave is not the same as the theoretical value,

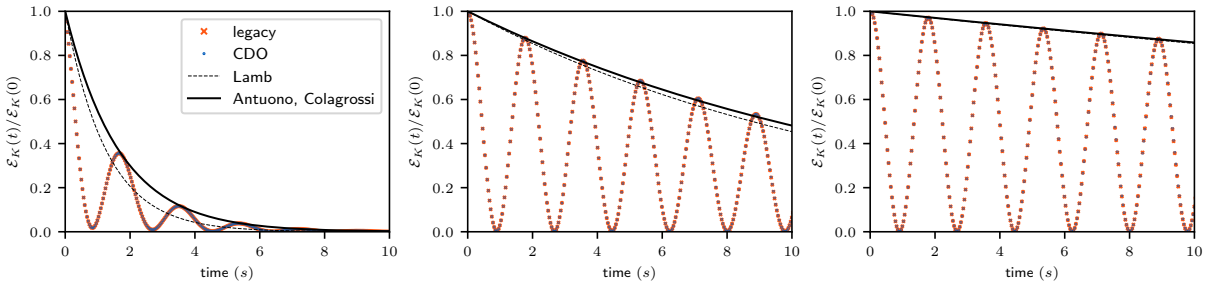


Figure 10: Evolution of the kinetic energy for $Re = 50$ (left), $Re = 500$ (center), and $Re = 2500$ (right), showing the decay of energy as compared to the theoretical solution from Lamb [34] and Antuono and Colagrossi [1].

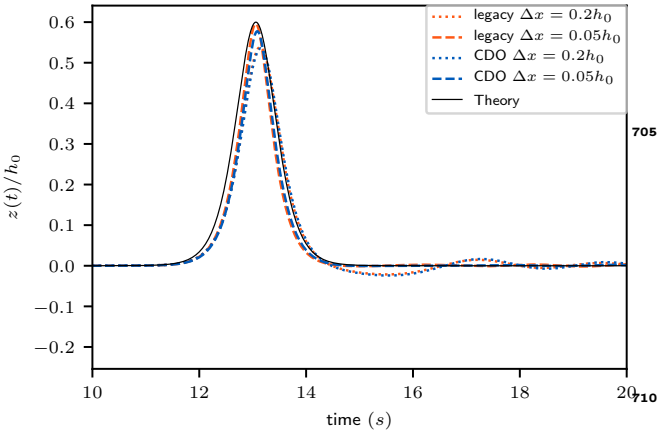


Figure 11: Wave elevation at $x = 20\,\mathrm{m}$ over time for different discretizations (dotted: $\Delta x = \Delta z = 0.2h_0$; dashed: $\Delta x = \Delta z = 0.05h_0$), compared to the theoretical solution (solid line), using the legacy solver.

taken to be the convective outlet velocity).

In order to understand the errors associated with wave propagation, we consider three different grids, with 200×5 cells ($\Delta t = 0.01\sqrt{\frac{h_0}{g}}$), 400×10 cells ($\Delta t = 0.005\sqrt{\frac{h_0}{g}}$), and 800×20 cells ($\Delta t = 0.0025\sqrt{\frac{h_0}{g}}$). With $\Delta x = \Delta z = 0.1h_0$, we get an arrival time (and thus wave-speed) error of less than 1% for both legacy and CDO solvers.

We also see a near linear convergence rate for the wave amplitude (Table 2), which is expected for the low-order method used here.

4.5. Wave propagation over a submerged bar

A common test of non-hydrostatic wave models is propagation over a submerged bar, which is beyond the range of a shallow-water assumption, or even some Boussinesq models (*e.g.*, the $O(\mu)^2$ model of [50]). We consider here the experimental results of [4], comparing against time-series measured at various wave gauges.

In this test case, 17500 hexahedral cells are used to discretize the domain which is physically 30 m long, with a depth of 0.4 m, with a bar of minimum depth 0.1 m. For this case, a $k - \epsilon$ turbulence model will be used, in part to demonstrate that it does not negatively impact the propagation of the waves in non-breaking conditions.

Waves are absorbed on the right side through a surface pressure proportional to the vertical velocity at the surface, similar to [25]. Here, we impose a surface absorbing pressure, $p_{abs}(x,t)$ on the free-surface, which smoothly increases from zero inside the domain, up to some maximum, depending on the vertical velocity, u_z, at the free-surface:

$$p_{abs}(x,t) = \begin{cases} C_{AB}(\rho u_z \sqrt{gh}) \left(\frac{x - x_{AB}}{h}\right)^2 & x > x_{AB} \\ 0 & x \leq x_{AB} \end{cases}, \quad (50)$$

where here $x_{AB} = 20\,\mathrm{m}$ and $C_{AB} = 0.003$ (chosen by trial-and-error).

On the left side, the velocity is imposed as a boundary condition, using second-order Stokes wave theory, similar to Ma et al. [38], to ensure mass conservation, with a wave period of 2.5 s. Two different test conditions are considered; one for non-breaking waves, and one with plunging waves.

Scheme	$\frac{\Delta x}{h_0} = \frac{\Delta z}{h_0}$	Crest	Rel. error	Arrival time	Rel. error	Reflection	Rel. error
Legacy	1/5	0.536	-10.75%	16.31	1.94%	0.064	10.67%
	1/10	0.573	-4.51%	16.09	0.53%	0.062	10.36%
	1/20	0.594	-1.05%	16.00	0.01%	0.064	10.58%
CDO	1/5	0.540	-9.93%	16.28	1.75%	0.037	6.25%
	1/10	0.570	-5.00%	16.10	0.62%	0.037	6.22%
	1/20	0.583	-2.92%	16.06	0.34%	0.045	7.52%

Table 2
Errors in wave-elevation at $x = 20\,\mathrm{m}$ for solitary wave propagation test case at different grid resolutions.

The incident wave parameters were chosen to best represent the signal recorded at the first wave gauge, at the toe of the submerged bar, in order to best match experimental conditions and imprecision in the physical wavemaker.

For the non-breaking condition, a 2.9 cm incident wave-height was considered in the basin. Looking at the wave elevation over time at both this wave-gauge (WG1, at $x = 6\,\mathrm{m}$), as well as at the front and back of the shallowest region of the submerged bar (WG3 and WG5, at $x = 12\,\mathrm{m}$ and $x = 14\,\mathrm{m}$), we see that the waves are well-represented (Fig. 12). For this case, the differences between legacy and CDO solvers is negligible.

For the plunging wave condition, 5.4 cm incident wave is considered for the same setup. Like the previous case, we are able to verify that the incident wave elevation at the toe of the bar is well represented by the code, and while the peak elevation is overpredicted, the overall form of the time-series is also captured over the bar (Fig. 13). Initially this is unexpected, as with the ALE approach we do not consider an overturning wave, but others [*e.g.*, 53, 9, 38] have observed that non-hydrostatic models are able to reproduce correctly some aspects of breaking waves. Notably, we see here that the turbulent kinetic energy and dissipation rate show that the turbulence model is most active exactly in the region where breaking waves are expected (Fig. 14), though clearly the exact dynamics of the free-surface are not captured.

Note that a homogeneous Neumann condition is used at

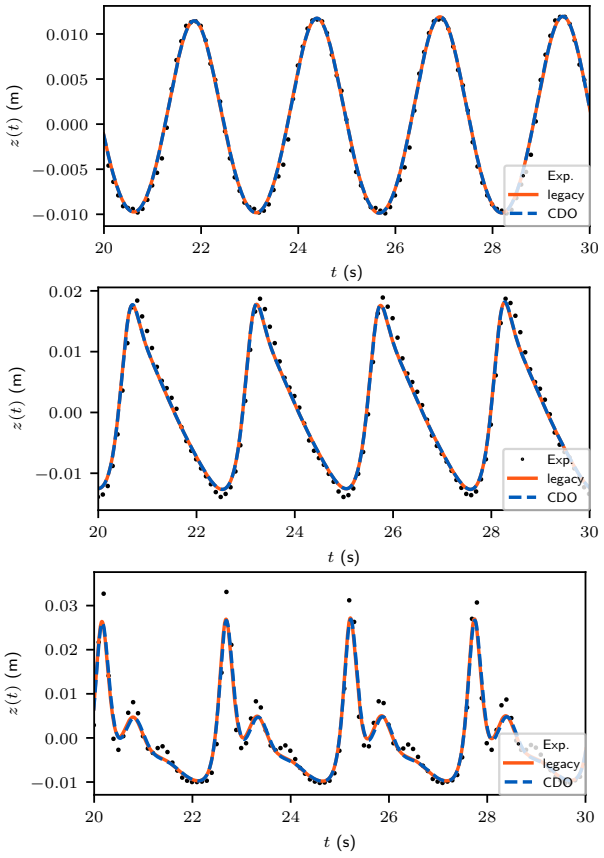


Figure 12: Wave elevation time-series at the toe of a submerged bar (at $x = 6\,\mathrm{m}$) and at the front (at $x = 12\,\mathrm{m}$) and back (at $x = 14\,\mathrm{m}$) of the top of the bar for non-breaking wave conditions, compared to Beji and Battjes [4].

the free-surface, but as the scale of the vortices (and thus turbulent viscosity) is reduced by the effect of the free-surface, in general a more sophisticated free-surface boundary condition is needed for the turbulent quantities [see 11, for more details].

One critique of general-purpose CFD models for the applications being considered here may be their computational

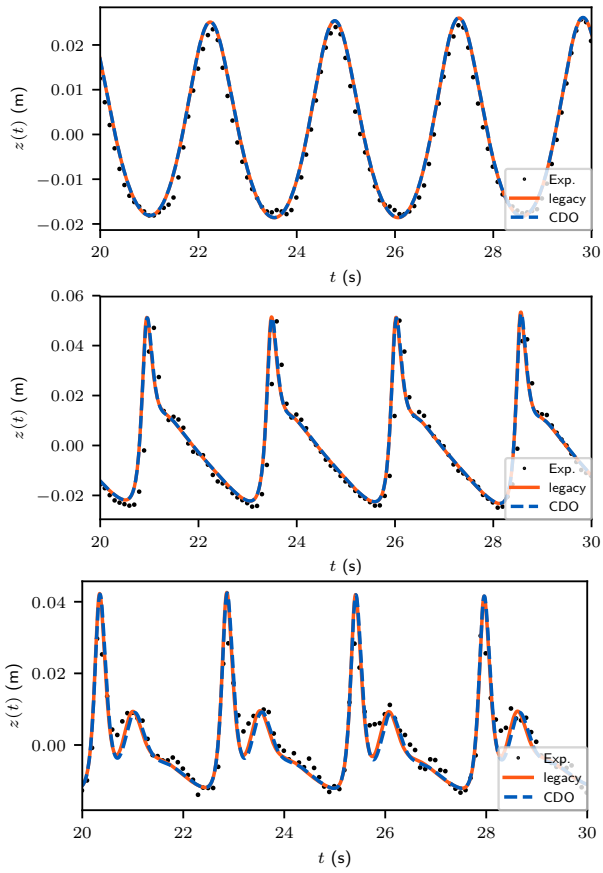


Figure 13: Wave elevation time-series at the toe of a submerged bar (at $x = 6\,\mathrm{m}$) and at the front (at $x = 12\,\mathrm{m}$) and back (at $x = 14\,\mathrm{m}$) of the top of the bar for plunging wave conditions, compared to Beji and Battjes [4].

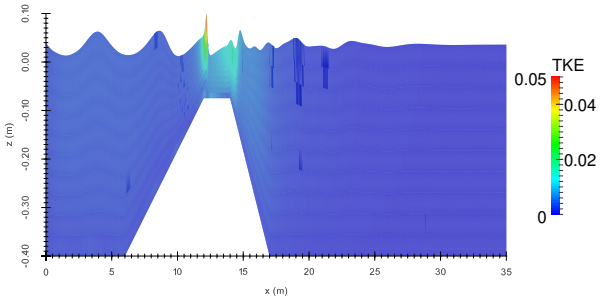


Figure 14: Turbulent kinetic energy ($\mathrm{m}^2/\mathrm{s}^2$) over a submerged bar under plunging wave conditions, corresponding to the experimental test case of Beji and Battjes [4], at $t = 31.2594\,\mathrm{s}$.

speed in comparison with specialized wave propagation models. As the submerged bar test case has frequently been considered for validation and demonstrating the dispersive properties of wave models, this can be used as a basis of comparison. For this we use the 3-D code NHWAVE [38], with an identical grid and identical starting timestep of $1/256^{th}$ of one wave period. NHWAVE uses an adaptive timestep in function of CFL number, but this does not vary significantly during this test case, so the two models should require a similar amount of computational effort. To minimize variations between the tests, results were obtained on a single Intel Core `i7` processor, with 645.0 s for *Code_Saturne* with the legacy scheme of *Code_Saturne* (and 2680.0 s with the unoptimized conjugate gradient linear solvers for CDO scheme) versus 559.0 s for NHWAVE. This is not meant to be an exact comparison between the two codes, as NHWAVE can benefit from higher-order approaches which are not present in *Code_Saturne*, and *Code_Saturne* can be used on types of grids which are not available with NHWAVE, but merely to show that *Code_Saturne* can be used to provide a reasonable performance.

4.6. Forces on a vertical cylinder in regular waves

A common application of wave-models is the understanding of forces on bodies. Here we consider the nonlinear forces exerted on a vertical bottom-mounted cylinder (*i.e.*, a monopile) in regular waves. Huseby and Grue [31] conducted a range of experiments which has commonly been used for benchmarking numerical tools [23, 44] for the forces of periodic forces on a cylinder of radius $R = 3\,\mathrm{cm}$, in a depth of 0.6 m, at a range of wave conditions. Here we consider the case with $kR = 0.245$, and a wave steepness of $kA = 0.10$. For this application, no turbulence model was used, but a viscosity of $1 \times 10^{-5}\,\mathrm{m}^2/\mathrm{s}$ was applied, slightly higher than the molecular viscosity, in order to stabilize the model and account for some viscous effects.

The grid used consists of 77406 hexahedral cells in a domain of width 0.5 m and length 3.5 m, with the cylinder positioned in the center of the domain, with a time-step of $T/100$. Second-order Stokes waves are used to impose the incoming waves, and the same damping setup as the pre-

	Code_Saturne	Exp.
$\lvert F_1\rvert/(\rho g A R^2)$	6.42	6.45
$\lvert F_2\rvert/(\rho g A^2 R)$	0.41	0.40
$\lvert F_3\rvert/(\rho g A^3)$	0.29	0.37

Table 3
Harmonic components of the surge force on a monopile in deep water, with $kR = 0.245$ and wave amplitude $kA = 0.1$, comparing with the experiments of Huseby and Grue [31].

vious case (Eq. 50) is re-used here, with $x_{damp} = 2.5\,\mathrm{m}$ and $C_{damp} = 0.25$. The simulations are performed for 10 wave periods T, with a quasi-steady state being reached for $t > 7T$. A short-time Fourier transform is used to evaluate the convergence of the integrated force on the cylinder in the x-direction, $f(t)$ (Fig. 15):

$$F_{(m)}(t) = \frac{2}{T}\int_t^{t+T} F(t)\, e^{im\omega\tau}\,\mathrm{d}\tau, \tag{51}$$

and we see that the computed force compares well to the experimentally determined values (Table 3), using data from four periods, after a spin-up time of 7 periods has passed.

Further study of different amplitudes showed that local instabilities near the monopile are found at higher incident wave-heights, likely due to local wave breaking or an importance of local viscous effects that require a finer mesh than considered here. Here the legacy solver was used for the presented calculations, though little change was seen with the CDO solver.

4.7. Bottom-tilting wave-maker

In the ocean, tsunamis exhibit extremely long wavelengths that can be difficult to realistically replicate in a physical wave tank or basin, which are normally designed to handle conditions representative of wind-waves. While there has been extensive studies with solitary waves, it is known that solitary waves do not correctly represent the waves which

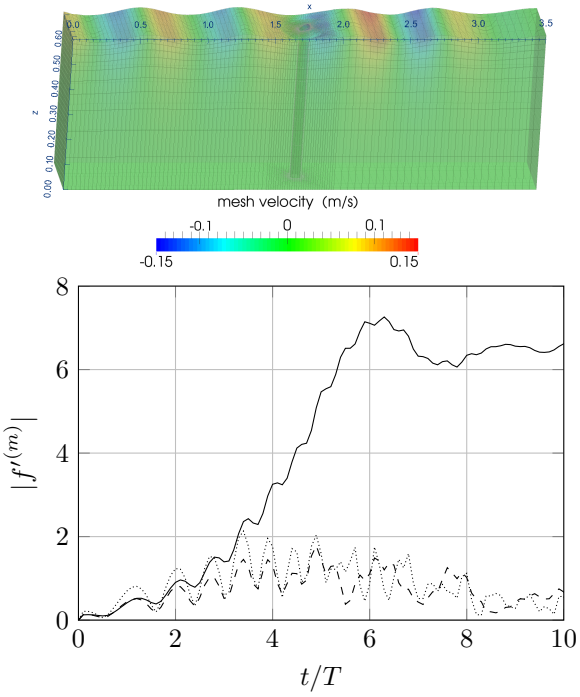


Figure 15: A vertical cylinder in regular waves (upper panel, at $t = 10T$), and the corresponding variation in time of the first three harmonics of the horizontal force, as evaluated by a short-time Fourier transform.

are actually experienced in a tsunami, as shown by Madsen et al. [40]. As a result, Lu et al. [37] has developed a bottom-tilting wave maker, testing on a small wave tank, 2 m long (see Fig. 16a; *i.e.*, $L = 1\,\mathrm{m}$). This provides a validation case where one of the boundaries is thus moving.

This 1 m long flap is either starting from a lowered position, moving up, or starting from a raised position, moving down, and stopping when the bed is flat. One can then propagate the waves over a long time, reflecting between two walls, even from a very small tank. Numerically, this is described such that the depth, $h(x,t)$, is:

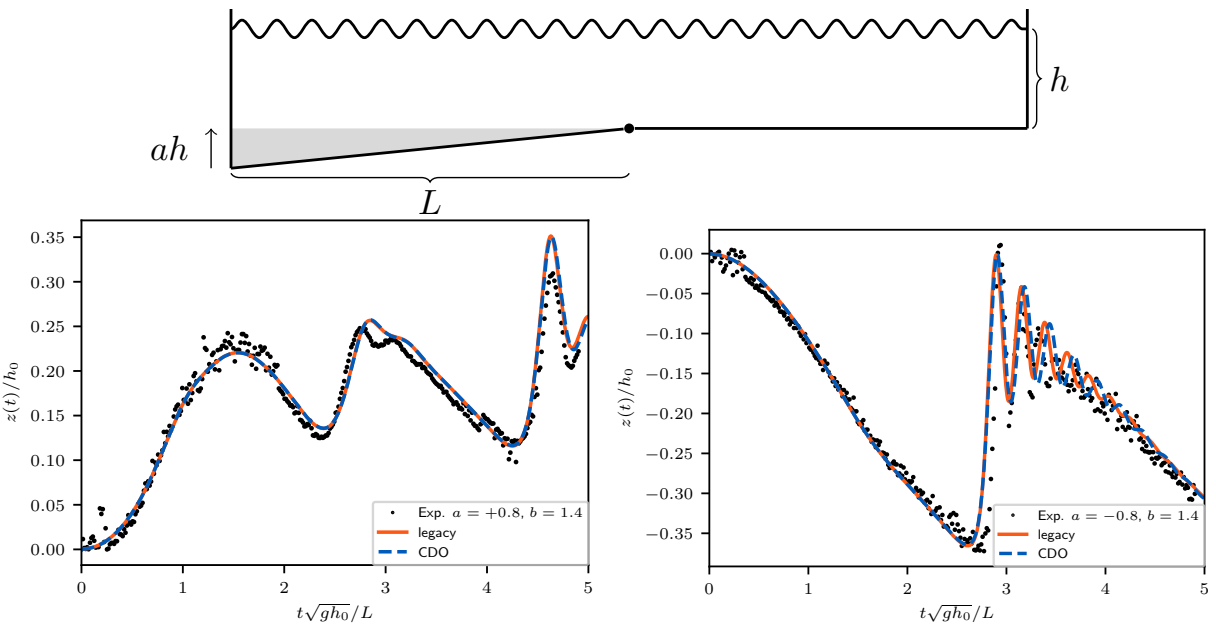


Figure 16: Bottom-tilting wave-maker schematic (top panel), and the wave profiles that are generated by an initial movement, compared to the experiments of [37].

$$h(x,t) = \begin{cases} h_0 & x > L \\ & \text{or } t > \frac{bL}{\sqrt{gh_0}}, \\ h_0 + \frac{L-x}{L}(ah_0)[1 - \frac{t\sqrt{gh_0}}{Lb}] & x < L \\ & \text{and } t < \frac{bL}{\sqrt{gh_0}}, \end{cases} \tag{52}$$

where the prescribed motion has a non-dimensional amplitude a and time b. For validation, we consider the results for a resting water depth of h_0=5 cm, with a duration $b = 1.4$, and amplitude $a = \pm 0.8$.

The grid is discretized with 5 vertical levels and 100 cells in the lengthwise direction and 11 cells in the spanwise direction (*i.e.*, $\Delta x = 2$ cm, $\Delta y = 1$ cm, and $\Delta z \approx 1$ cm), and a time-step of 0.009 s is used. It is assumed that all boundaries are smooth walls. The flap motion itself is imposed by fixing the mesh velocity.

The wave elevation was measured at a gauge at the center of the tank (*i.e.*, directly over the hinge), and we are able to reproduce the experimental values to a reasonable degree of accuracy, for both upward (Fig. 16b) and downward (Fig. 16c) motion.

5. Summary

Here we present recent developments of the ALE module of *Code_Saturne* for the simulation of free-surface flows, and then the results of applying the resulting model to ocean waves. Although certainly there are limitations to using a finite volume ALE model such as *Code_Saturne* in term of wave steepness, we can see that for certain circumstances, understanding the limitations of numerical dissipation and lower-order elements, a general CFD solver not specialized for water wave propagation can produce useful results in reasonable amount of time. Validation tests were presented for multiple setups, showing that the model is able to exactly conserve mass with the vertex-based solver on the dual barycentric mesh, correctly conserve energy, propagate waves, and compute forces on structures.

In future work, comparisons will be made between the

ALE module and the recently released VOF module, which will enable the complex modeling of breaking waves, but at an increased computational cost. In order to offset this increase in computational time, a coupling between *Code_Saturne* and far-field inviscid models will be considered.

Acknowledgements

We would like to thank Michel Benoit and Damien Violeau for their helpful remarks and discussions, Jérôme Bonelle for the early access to CDO schemes for the mesh velocity Poisson equation, and Yong-Sung Park and Heng Lu for data regarding the bottom-tilting wavemaker.

References

[1] Antuono, M., Colagrossi, A., 2013. The damping of viscous gravity waves. Wave Motion 50, 197–209.

[2] Archambeau, F., Méchitoua, N., Sakiz, M., 2004. Code_saturne: a finite volume method for the computation of turbulent incompressible flows - industrial applications. International Journal on Finite Volumes 1, 1–62.

[3] Bahlali, M.L., Henry, C., Carissimo, B., 2020. On the well-mixed condition and consistency issues in hybrid Eulerian/Lagrangian stochastic models of dispersion. Boundary-Layer Meteorology 174, 275–296.

[4] Beji, S., Battjes, J.A., 1993. Experimental investigation of wave-propagation over a bar. Coastal Engineering 19, 151–162.

[5] Benoit, M., Dias, F., Herterich, J., Scolan, Y.M., 2018. Un cas-test discriminant pour la simulation de la propagation et du run-up de trains de vagues de type tsunami, in: Proceedings of 16th Journées de l'Hydrodynamique, p. 14.

[6] Benson, D.J., 1989. An efficient, accurate, simple ALE method fo rnonlinear finite element programs. Computer Methods in Applied Mechanics and Engineering 72, 305–350.

[7] Bonelle, J., Ern, A., 2014. Analysis of Compatible Discrete Operator schemes for elliptic problems on polyhedral meshes. ESAIM Mathematical Modelling and Numerical Analysis 48, 553–581.

[8] Boscheri, W., Balsara, D.S., Dumbser, M., 2014. Lagrangian ADER-WENO finite volume schemes on unstructured triangular meshes based on genuinely multidimensional HLL Riemann solvers. Journal of Computational Physics 267, 112–138.

[9] Bradford, S.F., . Nonhydrostatic model for surf zone simulation. J. Waterway Port Coastal Ocean Eng. 137, 163–174.

[10] Carré, G., Del Pino, S., Després, B., Labourasse, E., 2009. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. Journal of Computational Physics 228, 5160–5183.

[11] Celik, I., Rodi, W., 1984. Simulation of free-surface effects in turbulent channel flows. Physiochemical Hydrodynamics 5, 217–227.

[12] Chan, R., Street, R., 1970. Computer study of finite-amplitude water waves. Journal of Computational Physics 6, 68–94.

[13] Chemartin, L., Lalande, P., Montreuil, E., Delalondre, C., Cheron, B., Lago, F., 2009. Three dimensional simulation of a DC free burning arc. Application to lightning physics. Atmospheric Research 91, 371–380.

[14] Chen, L., Zang, J., Hills, A., Morgan, G., Plummer, A., 2014. Numerical investigation of wave-structure interaction using OpenFOAM. Ocean Engineering 88, 91–109.

[15] Colas, C., Ferrand, M., Hérard, J.M., Latché, J.C., Le Coupanec, E., 2019. An Implicit Integral Formulation to Model Inviscid Fluid Flows in Obstructed Media. Computers & Fluids 188, 136–163.

[16] Dal Secco, S., Juan, O., Louis-Louisy, M., Lucas, J.Y., Plion, P., Porcheron, L., 2015. Using a genetic algorithm and CFD to identify low NOx configurations in an industrial boiler. Fuel 158, 672–683.

[17] Donea, J., Huerta, A., Ponthot, J.P., guez Ferran, A.R., 2004. Arbitrary Lagrangian-Eulerian Methods, in: Stein, E., de Borst, R., Hughes, T.J.R. (Eds.), Encyclopedia of Computational Mechanics. John Wiley & Sons. chapter 14, pp. 413–437.

[18] Durst, F., Maxworth, T., Pereira, J.C.F., 1989. Piston-driven, unsteady separation at a sudden expansion in a tube: Flow visualization and lda measurements. Physics of Fluids A 1, 1249.

[19] Dutykh, D., Clamond, D., 2014. Efficient computation of steady solitary gravity waves. Wave Motion 51, 86–99.

[20] Eymard, R., Gallouët, T., Guichard, C., Herbin, R., Masson, R., 2014. TP or not TP, that is the question. Computational Geosciences 18, 285–296. doi:10.1007/s10596-013-9392-9.

[21] Farhat, C., Geuzaine, P., Grandmont, C., 2001. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. Journal of Computational Physics 174, 669–694.

[22] Ferrand, M., Fontaine, J., Angelini, O., 2014. An anisotropic diffusion finite volume algorithm using a small stencil, in: Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pp. 577–585.

[23] Ferrant, P., 2000. Fully nonlinear interactions of long-crested wave packets with a three-dimensional body, in: 22nd Symposium on Naval Hydrodynamics, pp. 403–416.

[24] Gauffre, M.C., Benhamadouche, S., Badel, P.B., 2020. Wall-Modeled Large Eddy Simulation of the Flow Through PWR Fuel Assemblies at $Re_H = 66\,000$ - Validation on CALIFS Experimental Setup. Nuclear Technology 206, 255–265.

[25] Grilli, S.T., Horrillo, J., 1997. Numerical generation and absorption of fully nonlinear periodic waves. Journal of Engineering Mechanics 123, 1060–1069.

[26] Harlow, F.H., Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 8, 2182–2189.

[27] Helluy, Philippe, Golay, Frédéric, Caltagirone, Jean-Paul, Lubin, Pierre, Vincent, Stéphane, Drevard, Deborah, Marcer, Richard, Fraunié, Philippe, Seguin, Nicolas, Grilli, Stephan, Lesage, Anne-Cécile, Dervieux, Alain, Allain, Olivier, 2005. Numerical simulations of wave breaking. ESAIM: M2AN 39, 591–607. URL: https://doi.org/10.1051/m2an:2005024, doi:10.1051/m2an:2005024.

[28] Hervouet, J.M., 2007. Hydrodynamics of free surface flows: modelling with the finite element method. Wiley and Sons, Ltd.

[29] Hirt, C.W., Amsden, A.A., Cook, J.L., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics 14, 227–253.

[30] Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics 39, 201–225.

[31] Huseby, M., Grue, J., 2000. An experimental investigation of higher-harmonic wave forces on a vertical cylinder. Journal of Fluid Mechanics 414, 75–103.

[32] Issa, R.I., Gosman, A., Watkins, A., 1986. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics 62, 66–82.

[33] Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFOAM. International Journal for Numerical Methods in Fluids 70, 1073–1088.

[34] Lamb, H., 1945. Hydrodynamics. Dover Books.

[35] Lin, P., 2008. Numerical modeling of water waves. CRC Press.

[36] Longuet-Higgins, M.S., Cokelet, E., 1976. The deformation of steep surface waves on water – I. A numerical method of computation. Proceedings of the Royal Society A 350, 1–26.

[37] Lu, H., Park, Y.S., Cho, Y.S., 2017. Modeling of long waves generated by bottom-tilting wave maker. Coastal Engineering 122, 1–9.

[38] Ma, G., Shi, F., Kirby, J.T., 2012. Shock-capturing non-hydrostatic model for fully dispersive surface wave processes. Ocean Modelling 43-44, 22–35.

[39] Ma, L., Nmira, F., Consalvi, J., 2019. Verification and validation of a variable–density solver for fire safety applications. Numerical Heat Transfer, Part B: Fundamentals 76, 107–129.

[40] Madsen, P.A., Fuhrman, D.R., Schäffer, H.A., 2008. On the solitary wave paradigm for tsunamis. Journal of Geophysical Research 113, C12012.

[41] Mola, A., Heltai, L., DeSimone, A., 2013. A stable and adaptive semi-lagrangian potential model for unsteady and nonlinear ship-wave interactions. Engineering Analysis with Boundary Elements 37, 128 – 143. doi:https://doi.org/10.1016/j.enganabound.2012.09.005.

[42] Orlanski, I., 1976. A simple boundary condition for unbounded hyperbolic flows. Journal of Computational Physics 21, 251 – 269. doi:https://doi.org/10.1016/0021-9991(76)90023-1.

[43] Re, B., Dobryzynski, C., Guardone, A., 2017. An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. Journal of Computational Physics 340, 26–54.

[44] Shao, Y.L., Faltinsen, O.M., 2014. A harmonic polynomial cell (HPC) method for 3D Laplace equation with application in marine hydrodynamics. Journal of Computational Physics 274, 312–332.

[45] Storti, M.A., Garelli, L., Paz, R.R., 2012. A finite element formulation satisfying the discrete geometric conservation law based on averaged jacobians. International Journal for Numerical Methods in Fluids 69, 1872–1890. doi:10.1002/fld.2669.

[46] development team, C., 2019. *Code_Saturne* 6.0 Theory Guide. EDF R&D. URL: https://code-saturne.org/cms/sites/default/files/docs/6.0/theory.pdf.

[47] Teles, M.J., Pires-Silva, A.A., Benoit, M., 2013. Numerical modelling of wave current interactions at a local scale. Ocean Modelling 68, 72–87.

[48] Thomas, P.D., Lombard, C.K., 1979. Geometric conservation law and its application to flow computations on moving

grids. AIAA Journal 17, 1030–1037. doi:10.2514/3.61273, arXiv:https://doi.org/10.2514/3.61273.

[49] Versteeg, H.K., Malalasekera, W., 2007. An introduction to computational fluid dynamics the finite volume method. Second edition ed., Pearson Education Ltd., Harlow (GB) New York Boston [etc.]. URL: http://www.sudoc.fr/11762473X.

[50] Wei, G., Kirby, J.T., Grilli, S.T., Subramanya, R., 1995. A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves. Journal of Fluid Mechanics 294, 71–92.

[51] Wroniszewski, P.A., Verschaeve, J.C.G., Pedersen, G.K., 2014. Benchmarking of Navier-Stokes codes for free surface simulations by means of a solitary wave. Coastal Engineering 91, 1–17.

[52] Zhang, H., Reggio, M., Trépanier, J., Camarero, R., 1993. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. Computers & Fluids 22, 9 – 23. doi:https://doi.org/10.1016/0045-7930(93)90003-R.

[53] Zijlema, M., Stelling, G.S., 2008. Efficient computation of surf zone waves using the nonlinear shallow water equations with nonhydrostatic pressure. Coastal Engineering 55, 780–790.