



HAL
open science

The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows

Ting Song, Alex Main, Guglielmo Scovazzi, Mario Ricchiuto

► **To cite this version:**

Ting Song, Alex Main, Guglielmo Scovazzi, Mario Ricchiuto. The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows. [Research Report] RR-9136, Inria Bordeaux Sud-Ouest. 2017, pp.1-56. hal-01671990

HAL Id: hal-01671990

<https://hal.science/hal-01671990>

Submitted on 5 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows

T. Song , A. Main , G. Scovazzi, M. Ricchiuto

**RESEARCH
REPORT**

N° 9136

December 2017

Project-Team CARDAMOM



The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows

T. Song^{*}, A. Main[†], G. Scovazzi[‡], M. Ricchiuto[§]

Project-Team CARDAMOM

Research Report n° 9136 — December 2017 — 53 pages

Abstract: We propose a new computational approach for embedded boundary simulations of hyperbolic systems. Applications are shown for the linear wave equations and for the nonlinear shallow water system. The proposed approach belongs to the class of surrogate/approximate boundary algorithms and is based on the idea of shifting the location where boundary conditions are applied from the true to a surrogate boundary. Accordingly, boundary conditions, enforced weakly, are appropriately modified to preserve optimal error convergence rates. This framework is applied here in the setting of a stabilized finite element method, even though other spatial discretization techniques could have been employed. Accuracy, stability and robustness of the proposed method are tested by means of an extensive set of computational experiments for the acoustic wave propagation equations and shallow water equations. Comparisons with standard weak boundary conditions imposed on grids that conform to the geometry of the computational domain boundaries are also presented.

Key-words: Embedded boundary conditions; wave equation; shallow water flows; finite elements; approximate boundary methods

^{*} Department of Mechanical Engineering & Materials Science, Duke University, Durham, North Carolina 27708

[†] Department of Civil & Environmental Engineering and Department of Mechanical Engineering & Materials Science, Duke University, Durham, North Carolina 27708

[‡] Department of Civil & Environmental Engineering and Department of Mechanical Engineering & Materials Science, Duke University, Durham, North Carolina 27708

[§] Team CARDAMOM, INRIA Bordeaux Sud-Ouest, 200 av. de la vieille tour 33405 Talence Cedex, France

**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200, Avenue de la Vieille Tour
33405 Talence Cedex

Shifted boundary method pour systèmes hyperboliques: ondes linéaires et équations shallow water

Résumé : On propose une nouvelle approche pour des simulations avec bords immergés pour des systèmes hyperboliques et en particulier les équations shallow water. L'approche proposée consiste en modifier les conditions au bords avec un développement limité permettant d'assurer l'ordre deux avec des *embedded boundaries*. L'approche est implémenté est validée ici dans le cadre d'une méthode de type *stabilized finite element* sur un très grand nombre de cas tests représentatifs d'applications de propagation de vagues et inondation.

Mots-clés : conditions aux bords embedded; équations des ondes; équations shallow water; éléments finis; méthodes immergés

Contents

1	Introduction	3
2	The shallow water equations	6
2.1	Linearized shallow waters equations: Non-dissipative acoustic wave propagation	9
3	Preliminaries: General notation, the true domain, the surrogate domain and maps	11
4	A review of the weak enforcement of boundary conditions	13
4.1	Time-domain linear acoustics	13
4.2	Nonlinear shallow water equations	15
4.3	Discrete conservation properties	17
4.4	C-property: Preservation of a lake at rest	18
5	The Shifted boundary method	19
5.1	Acoustic waves	19
5.1.1	Dirichlet boundary conditions	19
5.1.2	Neumann boundary conditions	20
5.1.3	Variational formulation	20
5.2	Shallow water equations	21
5.3	Discrete conservation properties	24
5.4	C-property	24
6	Time integration	25
7	Numerical results for the wave propagation problem	25
7.1	Convergence tests	28
7.1.1	Neumann boundary condition on an embedded disk	28
7.1.2	Dirichlet boundary condition on an embedded disk	28
7.2	Acoustic waves around circle/rectangle	33
8	Numerical results for shallow water flows	34
8.1	C-property check	34
8.2	Convergence tests for shallow water flows inside a rectangular domain	35
8.2.1	Impermeable wall boundary conditions	37
8.2.2	River inflow boundary condition	37
8.2.3	River outflow boundary condition	38
8.3	Coastline test	43
8.4	Toce valley flash flood test: A simplified urban district layout	46

1 Introduction

In [31], the authors introduced an embedded method for the Poisson and Stokes problems, using an approximate (surrogate) domain representation and shifting the location where boundary conditions are imposed from the true to the surrogate boundary. In the present work, we extend those ideas to the case of hyperbolic systems and in particular the case of wave propagation problems in acoustics and shallow water flows.

Part of the motivation of this work is that time-domain acoustics in complex geometry still presents some computational challenges, and similarly for shallow water flows in the presence of very complex boundary/coastline features.

Immersed and embedded boundary methods offer advantages over geometrically conformal computational methods (i.e., methods in which numerical grids conform to the geometrical shape of the physical domains of interest) in that the task of mesh generation is greatly simplified. On the other hand, the enforcement of boundary conditions is more challenging in embedded methods, both from the mathematical and data structure perspectives.

In immersed boundary methods, the flow equations are discretized continuously both inside and outside the physical domain, and a smooth approximation to the Dirac delta function is introduced on the physical boundary, with the purpose of imposing the boundary conditions. Immersed boundary methods are older than embedded methods, dating to the seminal work of Peskin [36], and being applied in the finite element context by Boffi and Gastaldi [5]. It was observed in [14] that immersed methods can be interpreted in the context of variational formulations as penalty methods, and for this reason share with them some of the shortcomings.

In embedded methods, the equations are discretized and solved only on the physical domain, and external regions are excluded from the computation. These methods generally do not have the problems mentioned for immersed boundary discretizations, as they employ a “sharp interface approximation.” Within the finite element context, this is typically done by means of weak enforcement of boundary conditions through Nitsche’s method [34] in combination with XFEM strategies, as a way to appropriately construct the solution’s approximation spaces [14].

In contrast to a simple penalty approach, Nitsche’s method provides a mechanism to weakly and consistently enforce boundary conditions and does not adversely affect the conditioning of the discretized problem. Unfortunately, the standard Nitsche-XFEM method suffers from instabilities on elements that are cut by the interface in such a way that only a small fraction of them remains inside the physical domain, and consequently their effective mesh size becomes extremely small. As a consequence, these areas of small support require an arbitrarily large penalty parameter, thereby destroying the good conditioning properties of Nitsche’s method. A similar challenge exists for embedded boundary methods of finite volume type, which can suffer from a similar though technically different version of the small cut cell problem.

A variety of clever techniques have been attempted to circumvent this difficulty [33, 3, 2], but in most cases only within the context of interior interface problems. For single material problems, Burman [7] introduced the *ghost penalty method*, in which the variational form is stabilized by introducing a penalization of the solution gradients at the interface separating cut and uncut elements. This method was applied to the Stokes problem in [8], to the Navier-Stokes equations in [45], and to two-phase flow in [44]. The ghost penalty method has some drawbacks, in that the introduction of a fourth order operator in the ghost penalty term may have a delicate implementation in the nonlinear case, may negatively affect the overall condition number, and increases the stencil size when using low order finite elements, with additional complications in the case of parallel computing. An alternative approach introduced for B-spline variational formulations, known as *Extended B-splines* [17, 18], involves eliminating via an extrapolation procedure those cut B-splines with small support. This technique was applied to the Navier-Stokes equations for moving boundary problems in [41] and [42].

One additional challenge for the classes of methods just mentioned is that they require the geometric construction of the partial elements cut by the embedded boundary,

typically a complex and computationally intensive process. Since some sort of adaptive quadrature is used [35, 12], it is often the case that a non-negligible portion of the overall wall-clock time for a simulation is spent handling the embedded boundary.

The small cut cell problem can be circumvented through the introduction of an *approximate domain* method, in which the true domain is replaced by a *surrogate domain*. Boundary conditions are imposed along the frontier of the surrogate domain, whose geometry is chosen to avoid cut cells. The challenge then reduces to designing effective (i.e., accurate and robust) boundary conditions on the surrogate boundary. One of the earliest approximate domain methods was proposed in [37] for inviscid multiphase compressible flow. Referred to as the *Ghost Fluid Method* (GFM), it was later applied to multiphase compressible flow in [13, 1], to compressible fluid-structure interaction in [19], and to multiphase fluid/structure interaction in [53]. An approximate domain approach has also been utilized for viscous incompressible flow in [21], where, however, the approximated domain concept is only harnessed for the continuity equation. Although lacking strong theoretical foundations, the Ghost Fluid Method proved very effective for the simulation of practical problems, due to its ease of implementation and avoidance of the small cut cell instability. Approximate domains methods were also explored in [11, 10] in the context of the shallow water equations.

In this work, we propose a new approach that falls in the category of embedded finite element methods and leverages a surrogate/approximate boundary strategy. The key feature of the proposed approach is the idea of *shifting* the location where the boundary conditions are applied from the true to the surrogate boundary, and, most importantly, of appropriately modifying the *shifted* boundary conditions in order to avoid a reduction in the convergence rates of the overall formulation. In fact, we show that if the boundary conditions originally applied to the true domain are not appropriately modified, only first-order convergence can be expected. The appropriate (modified) boundary conditions are then applied weakly, using, for example, a Nitsche strategy (other boundary enforcement strategies are also possible, e.g., Lagrange multipliers, etc.). This process yields a method which is simple, robust, accurate and efficient.

We apply our method to the shallow water equations, as a prototypical hyperbolic system. We also consider the linearized limit of the shallow water equations, constituted by the equations of acoustics [46, 50, 47, 20].

We implement the proposed approach for boundary conditions in the context of a stabilized finite element formulation, which takes inspiration specifically from [46, 50, 47] and more broadly from [15, 16, 51, 26, 22, 27, 28, 25, 24, 23, 29, 49, 40]. The proposed embedded method is not limited in applicability, however, to the specific stabilized method considered here. For example, the extension of the proposed approach to discontinuous Galerkin methods or residual redistribution schemes is straightforward.

The benefit of an embedded method in the context of shallow water flow is evident when considering the complex morphology of the ocean coastlines. In real scenarios, this geometric complexity may induce considerable meshing costs, which we attempt to significantly reduce with the proposed method. Applications of interest that we consider here are the treatment of complex coastlines as reflective walls in the framework of large scale simulations and the simulation of fine scale urban floods. In both cases, one may actually treat the coast and the constructions as part of the topography, but this will introduce hard requirements on the numerical method and the mesh generation. In fact, the alternative option of adapting the mesh to these geometries often requires very large numbers of cells, even when capturing the fine details of the flow may not be necessary. In this context our method has the benefit of relaxing the requirements on both the numerical scheme and the mesh, while allowing a full second order accuracy.

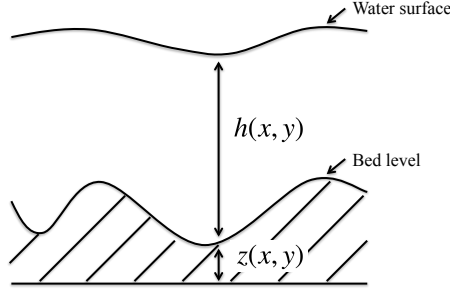


Figure 1: Definition of water height h and the bathymetry z .

We assess the accuracy, stability and robustness of the proposed method in a battery of tests, which also involve comparisons with finite element simulations on grids that conform to geometric boundaries.

2 The shallow water equations

The shallow water equations, also known as the *de Saint Venant* equations, are a simplified version of the Navier-Stokes equations for free-surface flows, and allow for the propagation of nonlinear waves. They are obtained by averaging the Navier-Stokes equations along the depth direction, and can be written as

$$\frac{\partial h}{\partial t} + \frac{\partial hv_1}{\partial x_1} + \frac{\partial hv_2}{\partial x_2} = 0, \quad (1a)$$

$$\frac{\partial}{\partial t} (hv_1) + \frac{\partial}{\partial x_1} \left(hv_1^2 + \frac{1}{2}gh^2 \right) + \frac{\partial}{\partial x_2} (hv_1v_2) = S_1, \quad (1b)$$

$$\frac{\partial}{\partial t} (hv_2) + \frac{\partial}{\partial x_1} (hv_1v_2) + \frac{\partial}{\partial x_2} \left(hv_2^2 + \frac{1}{2}gh^2 \right) = S_2. \quad (1c)$$

As illustrated in Figure 1, h is the height of the water column, z is the bathymetry of the water bed, free surface level $\eta = h + z$ and the two-dimensional position and velocity vectors are expressed in Cartesian coordinates as $\mathbf{x} = \{x_1, x_2\}$ and $\mathbf{v} = \{v_1, v_2\}$. $\partial/\partial t$ indicates derivation with respect to time, and

$$\mathbf{S} = \begin{Bmatrix} S_1 \\ S_2 \end{Bmatrix} = gh \begin{Bmatrix} S_{o1} - S_{f1} \\ S_{o2} - S_{f2} \end{Bmatrix}, \quad (2)$$

that is, \mathbf{S} is the source vector containing the slope of the basin (e.g., river bed, ocean floor) in the i th direction, $S_{oi} = -\frac{\partial z}{\partial x_i}$ and the friction $S_{fi} = \frac{f^2 v_i \sqrt{v_1^2 + v_2^2}}{h^{4/3}}$ is expressed through the Manning's roughness coefficient f . Additional forces can be added to the source term \mathbf{Z} if needed, to account, for example, for wind stress or other phenomena. The system (1a)–(1c) expresses the conservation of mass and momentum for a column of water of height h , is hyperbolic in structure and amenable to an analysis by means of the theory of characteristics and Riemann invariants. In the one-dimensional case, there are three characteristic speeds of propagation of information in the system, $v - c_s$, v , and $v + c_s$, where $c_s = \sqrt{gh}$ is the speed of gravity waves. Hence a signal can move with the velocity of the flow (in analogy to the entropy waves for the Euler equations of gas

dynamics), or can move with the speed associated with either one of two additional waves, one receding and the other advancing with respect to the flow velocity with the celerity of gravity waves, $\pm c_s$, respectively. Equations (1a)–(1c) can also be written in system vector form as

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{G} = \mathbf{Z}, \quad (3a)$$

where $\partial_t(\cdot) = \partial(\cdot)/\partial t$, for simplicity, and

$$\mathbf{U} = \begin{Bmatrix} h \\ hv \end{Bmatrix}, \quad \mathbf{G} = \mathbf{G}^v + \mathbf{G}^h, \quad \mathbf{G}^v = \mathbf{U} \otimes \mathbf{v}, \quad \mathbf{G}^h = \begin{Bmatrix} (\mathbf{0}_{2 \times 1})^T \\ 1/2gh^2 \mathbf{I}_{2 \times 2} \end{Bmatrix}, \quad \text{and} \quad \mathbf{Z} = \begin{Bmatrix} 0 \\ \mathbf{S} \end{Bmatrix}. \quad (3b)$$

Here $\mathbf{U} \otimes \mathbf{v} = \mathbf{U} \mathbf{v}^T$ and ∇ denotes the spatial gradient operator. The divergence operator $\nabla \cdot$, when expressed in Cartesian coordinates, is understood to apply to the second index of the 3×2 -matrix \mathbf{G} , that is $\nabla \cdot \mathbf{G} = \partial_{x_j} G_{ij}$. The system of equations (3a) is also often expressed as

$$\partial_t \mathbf{U} + \partial_{x_i} \mathbf{F}_i = \mathbf{Z}, \quad (4)$$

where the arrays \mathbf{F}_i are the columns of the matrix \mathbf{G} , and can be decomposed as

$$\mathbf{F}_i = \mathbf{F}_i^v + \mathbf{F}_i^h, \quad \mathbf{F}_i^v = v_i \mathbf{U}, \quad \mathbf{F}_i^h = \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \delta_i \end{Bmatrix}, \quad (5a)$$

with

$$\delta_i = \begin{Bmatrix} \delta_{1i} \\ \delta_{2i} \end{Bmatrix}. \quad (5b)$$

Using any set of variables \mathbf{X} , it is possible to rewrite the vector form (4) in quasi-linear form as

$$\mathbf{A}_0 \partial_t \mathbf{X} + \mathbf{A}_i \partial_{x_i} \mathbf{X} = \mathbf{Z}. \quad (6)$$

Here $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{X}}$ and $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{X}}$ is the i th Euler Jacobian matrix. When the vector of variables \mathbf{X} are the conservation variables \mathbf{U} , the Euler Jacobians become

$$\mathbf{A}_0 = \mathbf{I}_{3 \times 3}, \quad (7a)$$

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ -v_1^2 + gh & 2v_1 & 0 \\ -v_1 v_2 & v_2 & v_1 \end{bmatrix}, \quad (7b)$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 \\ -v_1 v_2 & v_2 & v_1 \\ -v_2^2 + gh & 0 & 2v_2 \end{bmatrix}. \quad (7c)$$

Another natural choice for the vector of variables \mathbf{X} are primitive variables $\mathbf{Y} = \{h, \mathbf{v}^T\}^T$, then the corresponding Euler Jacobians are

$$\hat{\mathbf{A}}_0 = \begin{bmatrix} 1 & 0 & 0 \\ v_1 & h & 0 \\ v_2 & 0 & h \end{bmatrix}, \quad (8a)$$

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} v_1 & h & 0 \\ v_1^2 + gh & 2hv_1 & 0 \\ v_1 v_2 & hv_2 & hv_1 \end{bmatrix}, \quad (8b)$$

$$\hat{\mathbf{A}}_2 = \begin{bmatrix} v_2 & 0 & h \\ v_1 v_2 & hv_2 & hv_1 \\ v_2^2 + gh & 0 & 2hv_2 \end{bmatrix}. \quad (8c)$$

Complete specification of the problem requires initial conditions on Ω , namely,

$$\mathbf{v}(t=0) = \mathbf{v}_0, \quad (9a)$$

$$h(t=0) = h_0, \quad (9b)$$

and appropriate boundary conditions. We consider some common boundary conditions for shallow water flows, which are best understood in the context of the theory of characteristics. The reader should not be confused by the fact that an open-sea boundary condition resembles a Dirichlet condition, and that a land boundary condition resembles a Neumann condition. All the proposed initial and boundary conditions can be derived uniquely using the theoretical framework of the Riemann invariants and generalized Riemann solvers. Here are a few typical examples:

Inviscid wall boundary conditions (similar to Neumann boundary conditions):

$$\mathbf{v} \cdot \mathbf{n} = 0, \quad \text{on } \Gamma_N, \quad (10)$$

which are typically applied when analyzing flooding around buildings in urban areas, and also when considering a coarse representation of the flow around large-scale coastal areas that are not the main focus of the simulation. The latter case is typical when there is interest in simulating a specific geographic area and it is necessary to also simulate some background, larger areas in order to appropriately characterize the dynamics of the overall flow, but these secondary areas are not of interest in the simulation, and only a coarse approximation is sufficient. Note that this type of boundary conditions conforms to the framework of the general Riemann problem, since the velocity is tangent to the boundary, and therefore only one of the characteristic lines goes across the boundary. In this case then, only one boundary condition is necessary, out of the possible three conditions that can be enforced (i.e., the height of the water column, the normal and tangential components of the velocity).

Open-sea (fixed height) boundary conditions (similar to Dirichlet boundary conditions):

$$h = \eta_D - z, \quad \text{on } \Gamma_D, \quad (11)$$

where the water height h is given by the free surface level η_D and the bathymetry z . These conditions are somewhat artificial, in the sense that a more appropriate boundary condition on the far field of an open domain should be given by non-reflective boundary conditions. These are in general considerably more complicated to implement, and can be constructed starting from the other boundary conditions mentioned here.

Subcritical river inflow boundary conditions ($|\mathbf{v} \cdot \mathbf{n}| \leq c_s$):

$$\begin{cases} h \mathbf{v} \cdot \mathbf{n} = m_{I;sub}, \\ \mathbf{v} \cdot \boldsymbol{\tau} = 0, \end{cases} \quad \text{on } \Gamma_{I;sub}, \quad (12)$$

which, as the name suggests, are typical of river inflows when the speed of propagation of gravity waves is faster than the normal velocity of the fluid at the inflow, that is $|\mathbf{v} \cdot \mathbf{n}| \leq c_s$. Recalling that $c_s = \sqrt{g h}$, it is easy to see that this situation occurs when the speed of the river inflow is relatively low with respect to the height of the water column. In this case there is always one characteristic line that exists the domain in the upstream

direction, and only two Riemann invariants (i.e. conditions) need to be imposed. The boundary condition on the water flux $q = h \mathbf{v} \cdot \mathbf{n}$ in (12) can be replaced by the alternative conditions

$$\mathbf{v} \cdot \mathbf{n} = v_{I;sub}, \quad \text{or} \quad h = \eta_{I;sub} - z, \quad \text{on } \Gamma_{I;sub}. \quad (13)$$

Supercritical river inflow boundary conditions ($|\mathbf{v} \cdot \mathbf{n}| > c_s$):

$$\begin{cases} \mathbf{v} = \mathbf{v}_{I;sup}, \\ h = \eta_{I;sup} - z, \end{cases} \quad \text{on } \Gamma_{I;sup}, \quad (14)$$

which are imposed at river inflows when the speed of propagation of gravity waves is slower than the normal velocity of the fluid at the inflow, that is $|\mathbf{v} \cdot \mathbf{n}| > c_s$. Opposite to the case of subcritical river inflow, this situation occurs for relatively high speed inflows, or relatively low water column heights. In this case no characteristic line can leave the domain upstream, and all three Riemann invariants of the system must be imposed, which means that we need three scalar conditions on the normal and tangential velocity components and the water height.

Subcritical river outflow boundary conditions ($|\mathbf{v} \cdot \mathbf{n}| \leq c_s$):

$$h \mathbf{v} \cdot \mathbf{n} = m_{O;sub}, \quad \text{on } \Gamma_{O;sub}. \quad (15)$$

This condition imposes only one constraint (the mass flow exiting the domain) on the three unknowns at the outflow. Alternatively, other less common boundary conditions can be imposed:

$$\mathbf{v} \cdot \mathbf{n} = v_{O;sub}, \quad \text{or} \quad h = \eta_{O;sub} - z, \quad \text{on } \Gamma_{O;sub}. \quad (16)$$

In this situation there are two characteristics that exit the outflow boundary and only one condition needs to be imposed, on the normal velocity, or the tangential velocity, or the water height.

Supercritical river outflow boundary conditions: In this case all characteristics exit the outflow, and no condition need to be imposed.

The boundary $\Gamma = \partial\Omega$ is partitioned as $\overline{\Gamma_D \cup \Gamma_N \cup \Gamma_{I;sub} \cup \Gamma_{O;sub} \cup \Gamma_{I;sup} \cup \Gamma_{O;sup}}$, in which all intersections (if any) of the partitions Γ_D , Γ_N , $\Gamma_{I;sub}$, $\Gamma_{O;sub}$, $\Gamma_{I;sup}$ and $\Gamma_{O;sup}$ are empty.

2.1 Linearized shallow waters equations: Non-dissipative acoustic wave propagation

In the limit of small disturbances \tilde{v} and \tilde{h} to the hydrostatic state of equilibrium of a fluid, denoted by $\bar{v} = 0$ and \bar{h} , we have

$$v_1 = \tilde{v}_1 \ll 1, \quad (17a)$$

$$v_2 = \tilde{v}_2 \ll 1, \quad (17b)$$

$$h = \bar{h} + \tilde{h}, \quad \bar{h} = \text{const.}, \quad \tilde{h} \ll 1. \quad (17c)$$

Substituting (17) in the shallow water equations (1a)-(1c) and neglecting higher-order terms, we obtain

$$\partial_t \tilde{h} + \nabla \cdot (\tilde{h} \tilde{\mathbf{v}}) = 0, \quad (18a)$$

$$\partial_t (\tilde{h} \tilde{\mathbf{v}}) + g \tilde{h} \nabla \tilde{h} = \mathbf{S}. \quad (18b)$$

Removing, for the sake of simplicity, the tilde above \mathbf{v} , defining $p = g\tilde{h}$, dividing the first and second equation by \tilde{h} , denoting by $c_s = \sqrt{\tilde{h}g}$ the speed of waves, $\chi = c_s^{-2}$ and by $\mathbf{b} = \tilde{h}^{-1} \mathbf{S}$, we have that (19) yield:

$$\chi \partial_t p + \nabla \cdot \mathbf{v} = 0, \quad (19a)$$

$$\partial_t \mathbf{v} + \nabla p = \mathbf{b}, \quad (19b)$$

which is the mixed system of equations governing the propagation of acoustic waves in a homogenous medium. Typical initial conditions are

$$\mathbf{v}(t=0) = \mathbf{v}_0, \quad (20a)$$

$$p(t=0) = p_0, \quad (20b)$$

and these complement appropriate pressure boundary conditions on the Dirichlet boundary Γ_D and normal velocity boundary conditions on the Neumann boundary Γ_N . Namely:

$$\mathbf{v} \cdot \mathbf{n}|_{\Gamma_N} = v_N(\mathbf{x}, t), \quad (21a)$$

$$p|_{\Gamma_D} = p_D(\mathbf{x}, t). \quad (21b)$$

Also (19) can be written in vector form. Denoting by $\mathbf{Y} = \{p, \mathbf{v}^T\}^T$ the solution state vector, we have

$$\partial_t \mathbf{U} + \partial_{x_i} \mathbf{F}_i = \mathbf{Z}, \quad \text{in } \Omega, \quad (22)$$

where

$$\mathbf{U} = \begin{Bmatrix} \chi p \\ \mathbf{v} \end{Bmatrix}, \quad \mathbf{Z} = \begin{Bmatrix} 0 \\ \mathbf{b} \end{Bmatrix}, \quad (23)$$

and the flux vector \mathbf{F}_i can be decomposed into two parts, that is, \mathbf{F}_i^v associated with the velocity component of the solution and \mathbf{F}_i^p associated with the pressure component of the solution:

$$\mathbf{F}_i = \mathbf{F}_i^v + \mathbf{F}_i^p, \quad (24a)$$

$$\mathbf{F}_i^v = v_i \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix}, \quad (24b)$$

$$\mathbf{F}_i^p = p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \vdots \\ \delta_{n_d i} \end{Bmatrix}. \quad (24c)$$

Note the similarities in structure between (24) and (5). In order to recover the formalism of the quasi-linear shallow water equations found in equation (6), we can write

$$\mathbf{A}_0 \partial_t \mathbf{Y} + \mathbf{A}_i \partial_{x_i} \mathbf{Y} = \mathbf{Z}, \quad (25)$$

where

$$\mathbf{A}_0 = \begin{bmatrix} \chi & \mathbf{0}_{1 \times n_d} \\ \mathbf{0}_{n_d \times 1} & \mathbf{I}_{n_d \times n_d} \end{bmatrix}, \quad (26a)$$

$$\begin{aligned} \mathbf{A}_i &= \mathbf{F}_{i, \mathbf{Y}} \\ &= \begin{bmatrix} 0 & \delta_{1i} & \cdots & \delta_{n_d i} \\ \delta_{1i} & & & \\ \vdots & & \mathbf{0}_{n_d \times n_d} & \\ \delta_{n_d i} & & & \end{bmatrix}. \end{aligned} \quad (26b)$$

Note that the i th Euler Jacobian matrix $\mathbf{A}_i \in \mathbb{R}^{(n_d+1) \times (n_d+1)}$ are constant and symmetric for $i = 0, \dots, n_d$.

3 Preliminaries: General notation, the true domain, the surrogate domain and maps

Consider the domain Ω , an open set in \mathbb{R}^{n_d} with Lipschitz boundary $\Gamma = \partial\Omega$, where n_d is the number of space dimensions. Let \mathcal{T}_h be a shape-regular triangulation (in the sense of Ciarlet) constituted by a family of non-overlapping n_d -partitions/elements Ω_e of Ω (e.g., triangles/quadrilaterals for $n_d = 2$ or tetrahedra/hexahedra for $n_d = 3$) such that $\bar{\Omega} = \bigcup_{e=1}^{n_{el}} \Omega_e$ (where n_{el} is the total number of elements). We denote by $h_e = h_e(\Omega_e)$ the diameter of element e and $h = \max_{\Omega_e \in \mathcal{T}_h} h_e$. Denoting by $\omega \subset \Omega$ a portion of Ω (e.g., an element domain Ω_e), and by γ a portion of Γ , we define with

$$(v, w)_\omega = \int_\omega v w \quad \text{and} \quad (\mathbf{v}, \mathbf{w})_\omega = \int_\omega \mathbf{v} \cdot \mathbf{w} \quad (27a)$$

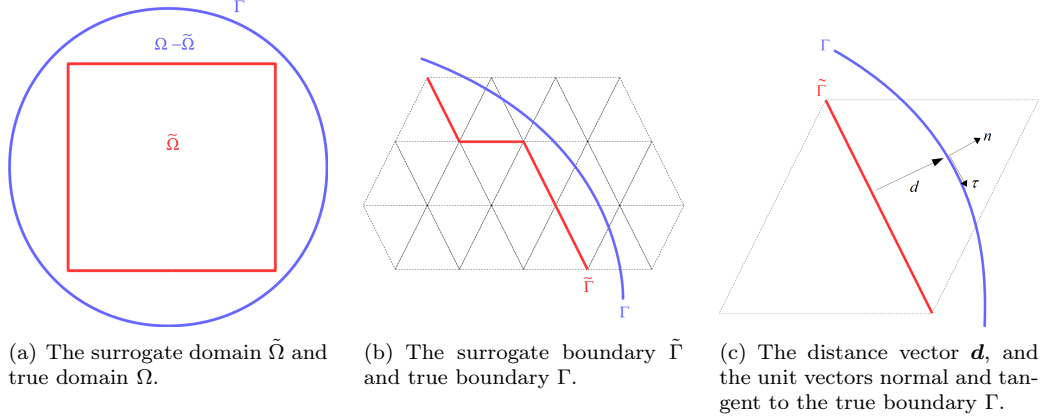
the $L^2(\omega)$ - and $(L^2(\omega))^{n_d}$ -inner products on the interior of ω and with

$$\langle v, w \rangle_\gamma = \int_\gamma v w \quad \text{and} \quad \langle \mathbf{v}, \mathbf{w} \rangle_\gamma = \int_\gamma \mathbf{v} \cdot \mathbf{w} \quad (27b)$$

boundary functionals on γ . In particular, the space $L^2(\mathcal{E})$ refers to the set of functions whose traces are square integrable on the interior or exterior boundary set \mathcal{E} . Let $\|v\|^2 = (v, v)_\omega$, and let $W_j^k(\Omega)$ be the Sobolev space with norm

$$\|v\|_{W_j^k(\Omega)} = \left(\sum_{\alpha \leq k} \|\mathbf{D}_x^\alpha v\|_{L^j(\Omega)}^j \right)^{1/j}, \quad 1 \leq j < \infty. \quad (28)$$

For $j = 2$, let $H^k(\Omega) = W_2^k(\Omega)$ and set $\|v\|_{W_j^k} = \|v\|_k$.

Figure 2: The surrogate domain, the boundary, and the vector \mathbf{d} .

Consider now an embedded discretization, in which the computational grid does not conform to the boundary, but is actually overlapped with it (see Fig. 2). The computational grid intersects the true boundary Γ of the domain Ω . We can then introduce a *surrogate boundary* $\tilde{\Gamma}$ composed of the edges/faces of the mesh that are the closest to the true boundary Γ . $\tilde{\Gamma}$ can be constructed, for example, by computing the intersections of the grid and the true boundary Γ and using closest-point projection algorithms to detect the closest face/edge of $\tilde{\Gamma}$ to Γ . Other choices are of course possible, as long as the overall topology of Γ and $\tilde{\Gamma}$ are close to each other, that is if Γ has a certain number of holes, the same number of holes needs to also be present also in $\tilde{\Gamma}$. To a certain extent, the construction of $\tilde{\Gamma}$ from Γ is a computational geometry problem, and several techniques can be borrowed from this field. The surrogate boundary $\tilde{\Gamma}$ encloses the *surrogate domain* $\tilde{\Omega}$. In particular, $\tilde{\mathbf{n}}$ indicates the unit outward-pointing normal to the surrogate boundary $\tilde{\Gamma}$, to be distinguished from the outward-pointing normal \mathbf{n} of Γ . We now define the map

$$\mathbf{M} : \tilde{\Gamma} \rightarrow \Gamma, \quad (29a)$$

$$\tilde{\mathbf{x}} \mapsto \mathbf{x}, \quad (29b)$$

which maps a point $\tilde{\mathbf{x}} \in \tilde{\Gamma}$ on the surrogate boundary to a point $\mathbf{x} \in \Gamma$ on the true boundary. For example, the map \mathbf{M} can also be built by means of the closest-point projection of points in $\tilde{\Gamma}$ onto Γ , as shown in Figure 2(c).

Remark 1. *Note that the closest-point projection, in spite of the segmented/faceted nature of the surrogate boundary $\tilde{\Gamma}$ is actually a smooth map from points in $\tilde{\Gamma}$ to points in Γ , and this is the primary reason why we propose it as the back-bone of the map \mathbf{M} .*

In particular, it will become very important to characterize the map \mathbf{M} through a distance vector function

$$\mathbf{d}_M(\tilde{\mathbf{x}}) = \mathbf{x} - \tilde{\mathbf{x}} = [\mathbf{M} - \mathbf{I}](\tilde{\mathbf{x}}). \quad (30)$$

In what follows, for the sake of simplicity, the subscript in the definition of \mathbf{d}_M will be omitted and we will simply write “ \mathbf{d} .” If the closest-point projection is used, the vector \mathbf{d} is aligned with \mathbf{n} . This choice is made throughout the rest of this article, and is stated as

Assumption 1. *The distance vector is defined as $\mathbf{d} = \|\mathbf{d}\|\mathbf{n}$, where the normal \mathbf{n} to the true boundary and the normal $\tilde{\mathbf{n}}$ to the surrogate boundary satisfy*

$$\mathbf{n} \cdot \tilde{\mathbf{n}} \geq 0, \quad (31)$$

The condition $\mathbf{n} \cdot \tilde{\mathbf{n}} \geq 0$ means that we require that $\tilde{\mathbf{n}}$ lies on the half-plane identified by the normal \mathbf{n} , a situation that is always verified in practice. Through the map \mathbf{M} , it is possible to define the extension $\bar{\psi}$ on $\tilde{\Gamma}$ of a function ψ initially defined only on Γ , as

$$\bar{\psi}(\tilde{\mathbf{x}}) \equiv \psi(\mathbf{M}(\tilde{\mathbf{x}})). \quad (32)$$

For example, the unit normal vector \mathbf{n} and unit tangential vectors $\boldsymbol{\tau}_i$ ($1 < i < n_d - 1$) to the boundary Γ can be extended to the boundary $\tilde{\Gamma}$ as follows:

$$\bar{\mathbf{n}}(\tilde{\mathbf{x}}) \equiv \mathbf{n}(\mathbf{M}(\tilde{\mathbf{x}})), \quad (33a)$$

$$\bar{\boldsymbol{\tau}}_i(\tilde{\mathbf{x}}) \equiv \boldsymbol{\tau}_i(\mathbf{M}(\tilde{\mathbf{x}})). \quad (33b)$$

In what follows, with the purpose of simplifying the notation, we will omit the bar from the expressions of $\bar{\mathbf{n}}$ and $\bar{\boldsymbol{\tau}}_i$, whenever this does not cause confusion. Therefore if, in the following, we write $\mathbf{n}(\tilde{\mathbf{x}})$ we actually mean $\bar{\mathbf{n}}(\tilde{\mathbf{x}})$, and similarly for $\boldsymbol{\tau}_i(\tilde{\mathbf{x}})$ and $\bar{\boldsymbol{\tau}}_i(\tilde{\mathbf{x}})$. We can also introduce the derivative of a function ψ along the directions $\bar{\mathbf{n}}$ and $\bar{\boldsymbol{\tau}}_i$ at a point $\tilde{\mathbf{x}} \in \tilde{\Gamma}$ as

$$\psi_{,\bar{\mathbf{n}}}(\tilde{\mathbf{x}}) = \nabla\psi(\tilde{\mathbf{x}}) \cdot \bar{\mathbf{n}}(\tilde{\mathbf{x}}) = \nabla\psi(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\mathbf{M}(\tilde{\mathbf{x}})), \quad (34a)$$

$$\psi_{,\bar{\boldsymbol{\tau}}_i}(\tilde{\mathbf{x}}) = \nabla\psi(\tilde{\mathbf{x}}) \cdot \bar{\boldsymbol{\tau}}_i(\tilde{\mathbf{x}}) = \nabla\psi(\tilde{\mathbf{x}}) \cdot \boldsymbol{\tau}_i(\mathbf{M}(\tilde{\mathbf{x}})). \quad (34b)$$

Observe also that the following Taylor expansion formula centered at $\tilde{\mathbf{x}} \in \tilde{\Gamma}$ holds for a generic field u at $\mathbf{x} = \mathbf{M}(\tilde{\mathbf{x}}) \in \Gamma$:

$$\begin{aligned} u(\mathbf{x}) &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot (\mathbf{x} - \tilde{\mathbf{x}}) + O(\|\mathbf{x} - \tilde{\mathbf{x}}\|^2) \\ &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot (\mathbf{M}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}) + O(\|\mathbf{M}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2) \\ &= u(\tilde{\mathbf{x}}) + \nabla u(\tilde{\mathbf{x}}) \cdot \mathbf{d}(\tilde{\mathbf{x}}) + O(\|\mathbf{d}(\tilde{\mathbf{x}})\|^2). \end{aligned} \quad (35)$$

The last expression in the chain of equalities can be used to develop a new strategy for the imposition of boundary conditions in the context of embedded methods. This approach is intrinsically only second-order accurate, unless additional terms in the Taylor expansion are included.

Remark 2. *Note that the closest-point projection, in spite of the segmented/faceted nature of the surrogate boundary $\tilde{\Gamma}$ is actually a smooth map from point in $\tilde{\Gamma}$ to points in Γ , and this is the primary reason why we propose it as the back-bone of the map \mathbf{M} .*

4 A review of the weak enforcement of boundary conditions

4.1 Time-domain linear acoustics

Let us first consider the case of weak enforcement of boundary conditions for conformal (i.e., not embedded) mesh computations of linear acoustic waves. This problem retains all the elements of the hyperbolic system of the shallow water equations, without the

inherent complexity of nonlinearities. The discussion that follows is a summary of the approach proposed in [50] for time domain acoustics, and its extension to the context of shallow water equations.

For the sake of clarity and without loss of generality, here we consider only the case of a semi-discrete formulation, in which the equations are discretized in space but not in time. We also consider solution and test spaces that are slightly more regular than the roughest possible case, that is, the following function spaces are chosen for the velocity and pressure

$$\mathcal{S}_p = \mathcal{V}_p = H^1(\Omega), \quad (36a)$$

$$\mathcal{S}_v = \mathcal{V}_v = H^{div}(\Omega). \quad (36b)$$

This last assumption is not a limitation, and the discussion can be extended to the most general case with relative ease. For example, the entire discussion can be adapted with minor changes to the case of discontinuous Galerkin approximation spaces. We consider a conforming finite element discretization, in which the discrete pressure and velocity spaces are subsets of the infinite dimensional ones above. For example, equal-order piecewise polynomial interpolation spaces can be used. Hence we write $\mathcal{S}_p^h \subset \mathcal{S}_p$, $\mathcal{V}_p^h \subset \mathcal{V}_p$, $\mathcal{S}_v \subset \mathcal{S}_v$ and $\mathcal{V}_v \subset \mathcal{V}_v$. Testing the strong form of the equation against the appropriate test functions and integrating by parts, we obtain:

Find $p \in \mathcal{S}_p^h$ and $\mathbf{v} \in \mathcal{S}_v^h$ such that, for all $\phi \in \mathcal{V}_p^h$ and $\boldsymbol{\psi} \in \mathcal{V}_v^h$,

$$0 = (\phi^h, \chi \partial_t p^h)_\Omega - (\nabla \phi, \mathbf{v}^h)_\Omega + \langle \phi^h, \widehat{\mathbf{v} \cdot \mathbf{n}} \rangle_\Gamma, \quad (37a)$$

$$0 = (\boldsymbol{\psi}^h, \partial_t \mathbf{v}^h - \mathbf{b})_\Omega - (\nabla \cdot \boldsymbol{\psi}^h, p^h)_\Omega + \langle \boldsymbol{\psi}^h \cdot \mathbf{n}, \hat{p} \rangle_\Gamma. \quad (37b)$$

The terms $\widehat{\mathbf{v} \cdot \mathbf{n}}$ and \hat{p} are numerical boundary traces of the solution and, if appropriately defined, can be used to impose boundary conditions weakly. The formulation we propose involves the following choices:

$$\widehat{\mathbf{v} \cdot \mathbf{n}} = \begin{cases} \mathbf{v} \cdot \mathbf{n} & \text{on } \Gamma_D, \\ v_N & \text{on } \Gamma_N, \end{cases} \quad (38a)$$

$$\hat{p} = \begin{cases} p_D & \text{on } \Gamma_D, \\ p & \text{on } \Gamma_N. \end{cases} \quad (38b)$$

Note that the normal velocity component is enforced directly on the boundary Γ_N , while the pressure is free, and the opposite happens on the Dirichlet boundary Γ_D . In what follows, we will omit, with the goal of a more readable notation, the superscript h on the components of the solution and the corresponding test functions, in spite of the fact that these fields are assumed fully discretized in space. The previous choice yields the final abstract formulation

$$0 = (\phi, \chi \partial_t p)_\Omega - (\nabla \phi, \mathbf{v})_\Omega + \langle \phi, v_N \rangle_{\Gamma_N} + \langle \phi, \mathbf{v} \cdot \mathbf{n} \rangle_{\Gamma_D}, \quad (39a)$$

$$0 = (\boldsymbol{\psi}, \partial_t \mathbf{v} - \mathbf{b})_\Omega - (\nabla \cdot \boldsymbol{\psi}, p)_\Omega + \langle \boldsymbol{\psi} \cdot \mathbf{n}, p_D \rangle_{\Gamma_D} + \langle \boldsymbol{\psi} \cdot \mathbf{n}, p \rangle_{\Gamma_N}, \quad (39b)$$

which can be rewritten in system vector form as

$$0 = (\mathbf{W}, \mathbf{A}_0 \partial_t \mathbf{Y} - \mathbf{Z})_\Omega - (\partial_{x_i} \mathbf{W}, \mathbf{F}_i)_\Omega + \langle \mathbf{W}, \widehat{\mathbf{F}_i \mathbf{n}_i} \rangle_\Gamma, \quad (40a)$$

with $\mathbf{W} = \{\phi, \psi^T\}^T$. We can then introduce the generalized vector \mathbf{H} of boundary conditions as follows:

$$\mathbf{H}(\mathbf{Y}) = \widehat{\mathbf{F}_i \mathbf{n}_i} = \begin{cases} v_N \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix} + p_D \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_D, \end{cases} \quad (40b)$$

which, recalling the identities

$$\mathbf{F}_i^p \mathbf{n}_i = p \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, \quad (41)$$

$$\mathbf{F}_i^v \mathbf{n}_i = \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix}, \quad (42)$$

can also be rewritten as $\mathbf{H} = \mathbf{H}^v + \mathbf{H}^p$, where

$$\mathbf{H}^v(\mathbf{Y}) = \begin{cases} v_N \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \mathbf{F}_i^v \mathbf{n}_i, & \text{on } \Gamma_D, \end{cases} \quad \mathbf{H}^p(\mathbf{Y}) = \begin{cases} \mathbf{F}_i^p \mathbf{n}_i, & \text{on } \Gamma_N, \\ p_D \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_D, \end{cases} \quad (43a)$$

a form that highlights more clearly the type of boundary condition enforced, and the corresponding data. Then the boundary conditions (21a)-(21b) can be recast as

$$\text{Neumann condition on } \Gamma_N : \mathbf{F}_i \mathbf{n}_i = \mathbf{H} \Leftrightarrow \mathbf{v} \cdot \mathbf{n} = v_N, \quad (44a)$$

$$\text{Dirichlet condition on } \Gamma_D : \mathbf{F}_i \mathbf{n}_i = \mathbf{H} \Leftrightarrow p \mathbf{n} = p_D \mathbf{n}. \quad (44b)$$

Hence, it is possible to reformulate (39a)-(39b) using a stabilized formulation, analogous to the one pursued in [46, 50, 47] in the context of hyperbolic wave systems:

$$0 = (\mathbf{W}, \mathbf{A}_0 \partial_t \mathbf{Y})_\Omega - (\partial_{x_i} \mathbf{W}, \mathbf{A}_i \mathbf{Y})_\Omega - (\mathbf{W}, \mathbf{Z})_\Omega + \langle \mathbf{W}, \mathbf{H} \rangle_\Gamma + (\mathcal{L} \mathbf{W}, \tau \mathbf{A}_0^{-1} (\mathcal{L}_t \mathbf{Y} - \mathbf{Z}))_\Omega, \quad (45)$$

where $\tau = c_\tau \frac{\Delta t}{2}$, and

$$\begin{aligned} \mathcal{L}_t &= \mathbf{A}_0 \partial_t + \mathcal{L}, \\ \mathcal{L} &= \mathbf{A}_i \partial_{x_i}. \end{aligned} \quad (46)$$

4.2 Nonlinear shallow water equations

In this case, boundary conditions can be enforced weakly with the following stabilized variational form, inspired again by the work in [46, 50, 47], and with close similarities to the variational forms developed in [15, 16]:

$$0 = (\mathbf{W}, \partial_t \mathbf{U} - \mathbf{Z})_\Omega - (\partial_{x_i} \mathbf{W}, \mathbf{F}_i)_\Omega + \langle \mathbf{W}, \mathbf{H} \rangle_\Gamma + (\mathcal{L} \mathbf{W}, \tau \hat{\mathbf{A}}_0 (\mathcal{L}_t \mathbf{Y} - \mathbf{Z}))_\Omega + (\partial_{x_i} \mathbf{W}, \nu g^{ij} \hat{\mathbf{A}}_0 \partial_{x_j} \mathbf{Y})_\Omega, \quad (47)$$

where again $\tau = c_\tau \frac{\Delta t}{2}$ is the stabilization characteristic time scale. The term $(\partial_{x_i} \mathbf{W}, \nu g^{ij} \hat{\mathbf{A}}_0 \partial_{x_j} \mathbf{Y})_\Omega$ in (47) is a *discontinuity capturing operator*, function of the artificial viscosity ν , which is defined as

$$\nu = \max \left(0, \left[\frac{(\mathcal{L}_t \mathbf{Y} - \mathbf{Z}) \cdot \hat{\mathbf{A}}_0^{-1} (\mathcal{L}_t \mathbf{Y} - \mathbf{Z})}{|\hat{\nabla}_\xi \mathbf{Y}|_{\hat{\mathbf{A}}_0}} \right]^{1/2} - \left| \frac{(\mathcal{L}_t \mathbf{Y} - \mathbf{Z}) \cdot \tau \hat{\mathbf{A}}_0 (\mathcal{L}_t \mathbf{Y} - \mathbf{Z})}{|\hat{\nabla}_\xi \mathbf{Y}|_{\hat{\mathbf{A}}_0}} \right| \right), \quad (48)$$

where $g^{ij} = [\xi_{k,i} \xi_{k,j}]^{-1}$, $\xi_k, k = 1, 2$, are the local element coordinates, with

$$|\hat{\nabla}_\xi \mathbf{Y}|_{\hat{\mathbf{A}}_0} = \mathbf{Y}_{,\xi_0} \cdot \hat{\mathbf{A}}_0 \mathbf{Y}_{,\xi_0} + g^{ij} \mathbf{Y}_{,i} \cdot \hat{\mathbf{A}}_0 \mathbf{Y}_{,j} \quad (49)$$

and the approximate time derivative in the element parent domain given as

$$\mathbf{Y}_{,\xi_0} = \frac{\mathbf{Y}(t_{n+1}) - \mathbf{Y}(t_n)}{2}. \quad (50)$$

See [15, 16] for more details. Note also that $\mathbf{H} = \mathbf{H}^v + \mathbf{H}^h$, and specifically,

$$\mathbf{H}^v(\mathbf{Y}) = \begin{cases} v_N \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_D, \\ m_{I;sub} \begin{Bmatrix} 1 \\ (\mathbf{v} \cdot \mathbf{n}) \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \mathbf{v}_{I;sup} \cdot \mathbf{n} \begin{Bmatrix} h_{I;sup} \\ h_{I;sup} \mathbf{v}_{I;sup} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ m_{O;sub} \begin{Bmatrix} 1 \\ \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}, \end{cases} \quad \mathbf{H}^h(\mathbf{Y}) = \begin{cases} \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \frac{1}{2} g h_D^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_D, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \frac{1}{2} g h_{I;sup}^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}, \end{cases} \quad (51)$$

$$\mathbf{H}^v(\mathbf{Y}) = \begin{cases} v_N \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_D, \\ m_{I;sub} \begin{Bmatrix} 1 \\ (\mathbf{v} \cdot \mathbf{n}) \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \mathbf{v}_{I;sup} \cdot \mathbf{n} \begin{Bmatrix} h_{I;sup} \\ h_{I;sup} \mathbf{v}_{I;sup} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ m_{O;sub} \begin{Bmatrix} 1 \\ \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \mathbf{v} \cdot \mathbf{n} \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}, \end{cases} \quad \mathbf{H}^h(\mathbf{Y}) = \begin{cases} \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \frac{1}{2}g(\eta_D - z)^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_D, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \frac{1}{2}g(\eta_{I;sup} - z)^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}. \end{cases} \quad (52)$$

Observe the following equivalences of boundary condition expressions:

$$\underbrace{\mathbf{H}(\mathbf{Y})^v + \mathbf{H}(\mathbf{Y})^h}_{\mathbf{H}(\mathbf{Y})} - \mathbf{F}_i n_i = \mathbf{0} \Leftrightarrow \begin{cases} (v_N - \mathbf{v} \cdot \mathbf{n}) \begin{Bmatrix} h \\ h \mathbf{v} \end{Bmatrix} = \mathbf{0} \Leftrightarrow \mathbf{v} \cdot \mathbf{n} = v_N, & \text{on } \Gamma_N, \\ \frac{1}{2}g \left((\eta_D - z)^2 - h^2 \right) \begin{Bmatrix} 0 \\ \mathbf{n} \end{Bmatrix} = \mathbf{0} \Leftrightarrow h = \eta_D - z, & \text{on } \Gamma_D, \\ h \mathbf{v} \cdot \mathbf{n} = m_{I;sub} \quad \text{and} \quad \mathbf{v} \cdot \boldsymbol{\tau} = 0, & \text{on } \Gamma_{I;sub}, \\ \mathbf{v} = \mathbf{v}_{I;sup} \quad \text{and} \quad h = \eta_{I;sup} - z, & \text{on } \Gamma_{I;sup}, \\ (m_{O;sub} - h \mathbf{v} \cdot \mathbf{n}) \begin{Bmatrix} 1 \\ \mathbf{v} \end{Bmatrix} = \mathbf{0} \Leftrightarrow h \mathbf{v} \cdot \mathbf{n} = m_{O;sub}, & \text{on } \Gamma_{O;sub}, \\ \mathbf{0} = \mathbf{0} \Leftrightarrow \text{No conditions imposed}, & \text{on } \Gamma_{O;sup}. \end{cases} \quad (53)$$

4.3 Discrete conservation properties

The variational formulation (47) naturally incorporates a conservation statement. Taking as test function $\mathbf{W} = \mathbf{1}_j$, where $\mathbf{1}_j$ is an array in which the j^{th} entry is unity and all the others are zero, we obtain:

$$0 = (\mathbf{1}_j, \partial_t \mathbf{U} - \mathbf{Z})_\Omega + \langle \mathbf{1}_j, \mathbf{H} \rangle_\Gamma, \quad (54)$$

which is a conservation statement of the mass and the various component of the momentum equation. In fact mass or momentum would not change if the integral of the boundary fluxes \mathbf{H} vanishes as well as the integral of the source vector \mathbf{Z} . A similar

statement can be obtained for the acoustic system (54) (recall that the pressure rate equation is effectively a statement of mass conservation).

4.4 C-property: Preservation of a lake at rest

When solving the shallow water equations, a main concern is the so-called *well-balanced* character of the discretization, which consists in the ability of the numerical method of *exactly* preserving the solution

$$\mathbf{v} = 0, \quad (55a)$$

$$h + z = \eta_0 \text{ (constant)}. \quad (55b)$$

This is the state of a flat free surface in stationary equilibrium and is often encountered as a background or initial state in many applications. It is important to exactly preserve the solution (55), since a small spurious perturbation can be amplified by the bathymetry (or analogous mechanisms). This property is often referred as the *C-property* (conservation property) and was originally introduced in [4].

The analysis of the C-property for the stabilized scheme proposed here is very similar to that of the residual based methods studied e.g. in [38]. We will proceed first by omitting the stabilization and discontinuity capturing terms from the discussion. These can be treated in a second stage. Substituting (55) in (47) as the solution leads to specific conditions on the discretization to maintain the C-property. Specifically, no inflow or outflow are possible, the integrals involving the term $\mathbf{F}_i^v = v_i \mathbf{U}$ vanish identically, as well as the volume integrals of the friction source term S_{fi} , as they involve variations of the flow speed. The only terms left are related to the hydrostatic pressure, and the bathymetry source of components $gh S_{oi} = -gh \partial z / \partial x_i$. In what follows, we will indicate with \mathbf{Z}_o the simplified version of \mathbf{Z} due to the previous simplifications. Now, if exact quadrature formulas are used for the evaluation of the terms $(\partial_{x_i} \mathbf{W}, \mathbf{F}_i^h)_\Omega$ and $(\mathbf{W}, \mathbf{Z}_o)_\Omega$, with respect to the piecewise linear interpolation spaces used in the approximation of h and z , then the continuity of the approximation and integration by parts over each element leads to

$$(\mathbf{W}, \partial_t \mathbf{U})_\Omega = (\mathbf{W}, \mathbf{Z}_o - \partial_{x_i} \mathbf{F}_i)_\Omega + \langle \mathbf{W}, \mathbf{F}_i^h n_i - \mathbf{H} \rangle_\Gamma. \quad (56)$$

Let us focus our attention now on the term $(\mathbf{W}, \mathbf{Z}_o - \partial_{x_i} \mathbf{F}_i)_\Omega$. Note first that the first component of $\mathbf{Z}_o - \partial_{x_i} \mathbf{F}_i$, corresponding to the mass conservation equation, vanish, and that the if ψ is a vector test function corresponding to the momentum equation, we have

$$(\psi_j, \partial_{x_j} (gh^2/2) + gh \partial_{x_j} z)_\Omega = (\psi_j, gh \partial_{x_j} (h + z))_\Omega = (\psi_j, gh \partial_{x_j} \eta_0)_\Omega = 0,$$

where the first equality holds because of the assumed exactness of the integration with respect to the linear variation of h within each mesh element, and the second is true as long as z and h are in the same space and (55) holds. We are thus left with the boundary term $\langle \mathbf{W}, \mathbf{F}_i^h n_i - \mathbf{H} \rangle_\Gamma$, which will cancel exactly.

Note now that in the case of a stabilized method with a discontinuity capturing operator, the additional terms are functions of the equation residuals, which vanish exactly in the case of the solution (55). Hence, we conclude that the stabilized variational form detailed in (47) satisfies the C-property.

Remark 3. *The above analysis is true as long as no dry areas are present in the domain. If there are elements in which $h = 0$ at some of the nodes, then the condition $\nabla(h +$*

$z) = \nabla\eta_0 = 0$ may be violated. This is due to the fact that the value of the polynomial interpolating the nodal values of $h + z$ may be different than η_0 at the dry node. To cure this issue, one must somehow modify the numerical approximation of the bathymetry to compensate for this unphysical effect. This is particularly necessary in the case in which the dry node is above the wet level in the element. A simple technique for this purpose is suggested in [39, 38] and consists in modifying the nodal values of the bathymetry as follows:

$$\hat{z}_j = \begin{cases} H_{\max} & \text{if } h_j = 0 \text{ and } z(\mathbf{x}_j) > H_{\max} \\ z(\mathbf{x}_j) & \text{otherwise} \end{cases}$$

where on each element

$$H_{\max} = \max_{\substack{k \in \text{element} \\ h_k > 0}} (h_k + z(\mathbf{x}_k))$$

Replacing the nodal values of z by \hat{z} in the computation of the integrals of bathymetric source terms allows to restore the C-property in dry cells. Some authors [6, 9] also suggest to couple this correction with a limiter on the mass flux in vicinity of dry areas.

5 The Shifted boundary method

While weak boundary conditions can be quite effective on conformal grids, their applicability in the case of embedded boundaries is more challenging, for the following reasons:

- a. The presence of cut elements of small size can render the overall approach numerically unstable and/or produce poor condition numbers in the ensuing algebraic system.
- b. Numerical integration on cut elements can be computationally expensive and/or difficult to implement.

Because both of these issues emanate from the mere existence of cut elements near the boundary, one idea could be to exclude them altogether from the simulation. Excluding these elements has the effect of moving the boundary Γ of the computational domain to the surrogate boundary $\tilde{\Gamma}$. Of course, if boundary conditions are naïvely applied on the surrogate boundary, then an $O(h)$ error is introduced.

We present next an embedded finite element formulation, which falls under the umbrella of the *shifted boundary* (SB) method [31, 32]. The key idea behind the SB method is *not* to apply boundary conditions on the true boundary Γ , but, rather, to *shift* their location to a surrogate boundary $\tilde{\Gamma}$. The map \mathbf{M} defined in (30) and the Taylor formula (35) are instrumental in imposing on $\tilde{\Gamma}$ a *shifted* boundary condition that is a second-order accurate approximation to the exact boundary condition on Γ .

5.1 Acoustic waves

We start from the hyperbolic system of non-dissipative acoustics described in Section 2.1 since it is a simpler prototype for the nonlinear shallow water equations.

5.1.1 Dirichlet boundary conditions

The goal of this section is to develop a suitable boundary condition on the surrogate Dirichlet boundary $\tilde{\Gamma}_D$ that, up to second-order accuracy, is equivalent to imposing the original boundary condition on the true boundary Γ_D . To achieve this, we perform the

following Taylor expansion of the pressure field along the direction $\mathbf{d} = \mathbf{M}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}$, from $\tilde{\mathbf{x}} \in \tilde{\Gamma}_D$ to $\mathbf{x} = \mathbf{M}(\tilde{\mathbf{x}}) \in \Gamma_D$,

$$\begin{aligned} 0 &\approx p(\tilde{\mathbf{x}}) + \nabla p(\tilde{\mathbf{x}}) \cdot (\mathbf{x} - \tilde{\mathbf{x}}) - p_D(\mathbf{x}) + O(\|\mathbf{x} - \tilde{\mathbf{x}}\|^2) \\ &= p(\tilde{\mathbf{x}}) + \nabla p(\tilde{\mathbf{x}}) \cdot (\mathbf{M}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}) - p_D(\mathbf{M}(\tilde{\mathbf{x}})) + O(\|\mathbf{M}(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2) \\ &= p(\tilde{\mathbf{x}}) + \nabla p(\tilde{\mathbf{x}}) \cdot \mathbf{d}(\tilde{\mathbf{x}}) - p_D(\mathbf{M}(\tilde{\mathbf{x}})) + O(\|\mathbf{d}(\tilde{\mathbf{x}})\|^2). \end{aligned} \quad (57)$$

The last expression in the chain of equalities can be used as the modified boundary condition on the surrogate boundary of Dirichlet type $\tilde{\Gamma}_D$, which preserves the accuracy of the true boundary condition up to second-order. Note that according to the notation that we introduced in Section 3 we could have written $\bar{p}_D(\tilde{\mathbf{x}}) = p_D(\mathbf{M}(\tilde{\mathbf{x}}))$. Then the shifted boundary condition on surrogate Dirichlet-type boundary $\tilde{\Gamma}_D$ reads

$$p|_{\tilde{\Gamma}_D} = \bar{p}_D(\tilde{\mathbf{x}}) - \nabla p(\tilde{\mathbf{x}}) \cdot \mathbf{d}(\tilde{\mathbf{x}}). \quad (58)$$

5.1.2 Neumann boundary conditions

A similar strategy can be elaborated to derive the boundary condition on the Neumann surrogate boundary $\tilde{\Gamma}_N$. The main difference with respect to the previous case of Dirichlet boundary conditions is that the normal $\tilde{\mathbf{n}}$ to the surrogate boundary and the normal \mathbf{n} to the true boundary do not coincide. This situation can be resolved by decomposing the unit normal vector $\tilde{\mathbf{n}}$ at $\tilde{\mathbf{x}}$ as $\tilde{\mathbf{n}} = (\tilde{\mathbf{n}} \cdot \mathbf{n}) \mathbf{n} + (\tilde{\mathbf{n}} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j$, where \mathbf{n} is the normal to Γ_N and $\boldsymbol{\tau}^j$ ($j = 1, \dots, n_d - 1$) are the vectors tangent to Γ_N , respectively, and we also recall that $\mathbf{n}(\tilde{\mathbf{x}}) = \mathbf{n}(\mathbf{M}(\tilde{\mathbf{x}}))$ and $\boldsymbol{\tau}^j(\tilde{\mathbf{x}}) = \boldsymbol{\tau}^j(\mathbf{M}(\tilde{\mathbf{x}}))$ by (33). Then we can apply the Taylor expansion to the velocity appearing in the term $\mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\tilde{\mathbf{x}})$, so that, again for the point $\tilde{\mathbf{x}}$ on the surrogate boundary $\tilde{\Gamma}_N$,

$$\begin{aligned} \mathbf{v}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) &= ((\mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}) \mathbf{n} + (\mathbf{v}(\tilde{\mathbf{x}}) \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j) \cdot \tilde{\mathbf{n}} \\ &\approx (((\mathbf{v}(\mathbf{x}) - \nabla \mathbf{v}(\tilde{\mathbf{x}}) \mathbf{d}) \cdot \mathbf{n}) \mathbf{n} + (\mathbf{v}(\tilde{\mathbf{x}}) \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j) \cdot \tilde{\mathbf{n}} \\ &= (v_N - \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + (\mathbf{v} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j \cdot \tilde{\mathbf{n}}, \end{aligned} \quad (59)$$

where $v_N = v_N(\mathbf{M}(\tilde{\mathbf{x}}))$. Equation (59) represents the surrogate Neumann boundary condition on $\tilde{\Gamma}_N$, and requires for consistency the tangential term $(\mathbf{v} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j \cdot \tilde{\mathbf{n}}$, a byproduct of the decomposition of $\tilde{\mathbf{n}}$ in terms of \mathbf{n} and $\boldsymbol{\tau}^j$.

5.1.3 Variational formulation

Using the approximate (shifted) Dirichlet and Neumann conditions we can modify formulation (37a)–(37b) as follows:

$$0 = (\phi, \chi \partial_t p)_{\tilde{\Omega}} - (\nabla \phi, \mathbf{v})_{\tilde{\Omega}} + \langle \phi, \widehat{\mathbf{v} \cdot \tilde{\mathbf{n}}} \rangle_{\tilde{\Gamma}}, \quad (60a)$$

$$0 = (\boldsymbol{\psi}, \partial_t \mathbf{v} - \mathbf{b})_{\tilde{\Omega}} - (\nabla \cdot \boldsymbol{\psi}, p)_{\tilde{\Omega}} + \langle \boldsymbol{\psi} \cdot \tilde{\mathbf{n}}, \hat{p} \rangle_{\tilde{\Gamma}}, \quad (60b)$$

where now

$$\widehat{\mathbf{v} \cdot \tilde{\mathbf{n}}} = \begin{cases} \mathbf{v} \cdot \tilde{\mathbf{n}}, & \text{on } \tilde{\Gamma}_D, \\ (v_N - \mathbf{n}^T (\nabla \mathbf{v}) \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + (\mathbf{v} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j \cdot \tilde{\mathbf{n}}, & \text{on } \tilde{\Gamma}_N, \end{cases} \quad (61a)$$

$$\hat{p} = \begin{cases} p_D - \nabla p \cdot \mathbf{d}, & \text{on } \tilde{\Gamma}_D, \\ p, & \text{on } \tilde{\Gamma}_N. \end{cases} \quad (61b)$$

Hence, upon substitution, we obtain the variational formulation

$$0 = (\phi, \chi \partial_t p)_{\tilde{\Omega}} - (\nabla \phi, \mathbf{v})_{\tilde{\Omega}} + \langle \phi, ((v_N - \mathbf{n}^T (\nabla \mathbf{v}) \mathbf{d}) \mathbf{n} + (\mathbf{v} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j) \cdot \tilde{\mathbf{n}} \rangle_{\tilde{\Gamma}_N} + \langle \phi, \mathbf{v} \cdot \tilde{\mathbf{n}} \rangle_{\tilde{\Gamma}_D}, \quad (62a)$$

$$0 = (\boldsymbol{\psi}, \partial_t \mathbf{v} - \mathbf{b})_{\tilde{\Omega}} - (\nabla \cdot \boldsymbol{\psi}, p)_{\tilde{\Omega}} + \langle \boldsymbol{\psi} \cdot \tilde{\mathbf{n}}, p_D - \nabla p \cdot \mathbf{d} \rangle_{\tilde{\Gamma}_D} + \langle \boldsymbol{\psi} \cdot \tilde{\mathbf{n}}, p \rangle_{\tilde{\Gamma}_N}. \quad (62b)$$

The previous equations can be compared with (39a)–(39b) to highlight differences with respect to the case of conformal grids. Integrating by parts in space leads to the Euler-Lagrange equations,

$$0 = (\phi, \chi \partial_t p + \nabla \cdot \mathbf{v})_{\tilde{\Omega}} - \langle (\tilde{\mathbf{n}} \cdot \mathbf{n}) \phi, \mathbf{v} + (\nabla \mathbf{v}) \mathbf{d} \cdot \mathbf{n} - v_N \rangle_{\tilde{\Gamma}_N}, \quad (63a)$$

$$0 = (\boldsymbol{\psi}, \partial_t \mathbf{v} + \nabla p - \mathbf{b})_{\tilde{\Omega}} - \langle \boldsymbol{\psi} \cdot \tilde{\mathbf{n}}, p - p_D + \nabla p \cdot \mathbf{d} \rangle_{\tilde{\Gamma}_D} \quad (63b)$$

which show that the SB method enforces the partial differential equations on the interior of surrogate domain $\tilde{\Omega}$ and the shifted boundary conditions on the surrogate boundary $\tilde{\Gamma}$. As in the case of conformal grid computations, also the variational SB formulation can be cast in vector form:

$$0 = (\mathbf{W}, \mathbf{A}_0 \partial_t \mathbf{Y} - \mathbf{Z})_{\tilde{\Omega}} - (\partial_{x_i} \mathbf{W}, \mathbf{A}_i \mathbf{Y})_{\tilde{\Omega}} + \langle \mathbf{W}, \widehat{\mathbf{F}}_i \tilde{\mathbf{n}}_i \rangle_{\tilde{\Gamma}}, \quad (64)$$

where

$$\widehat{\mathbf{F}}_i \tilde{\mathbf{n}}_i = \begin{cases} ((v_N - \mathbf{n}^T (\nabla \mathbf{v}) \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + (\mathbf{v} \cdot \boldsymbol{\tau}^j) \boldsymbol{\tau}^j \cdot \tilde{\mathbf{n}}) \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix} + \begin{Bmatrix} 0 \\ p \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \tilde{\Gamma}_N, \\ \mathbf{v} \cdot \tilde{\mathbf{n}} \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix} + (p_D - \nabla p \cdot \mathbf{d}) \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \tilde{\Gamma}_D. \end{cases} \quad (65)$$

Denoting $\tilde{\mathbf{H}} = \widehat{\mathbf{F}}_i \tilde{\mathbf{n}}_i$, with a little algebra, somewhat tedious but otherwise straightforward, we obtain

$$\tilde{\mathbf{H}} = \widehat{\mathbf{F}}_i \tilde{\mathbf{n}}_i = \begin{cases} \mathbf{n} \cdot \tilde{\mathbf{n}} \left(v_N \begin{Bmatrix} 1 \\ \mathbf{0}_{n_d \times 1} \end{Bmatrix} - n_i \mathbf{A}_i^v \nabla \mathbf{Y} \mathbf{d} \right) + \boldsymbol{\tau}^j \cdot \tilde{\mathbf{n}} \left(\mathbf{F}_i^v \boldsymbol{\tau}_i^j \right) + \mathbf{F}_i^p \tilde{\mathbf{n}}_i, & \text{on } \tilde{\Gamma}_N, \\ p_D \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix} - \tilde{n}_i \mathbf{A}_i^p \nabla \mathbf{Y} \mathbf{d} + \mathbf{F}_i^v \tilde{\mathbf{n}}_i, & \text{on } \tilde{\Gamma}_D, \end{cases} \quad (66)$$

in which $\mathbf{A}_i^v = \mathbf{F}_{i, \mathbf{Y}}^v$ and $\mathbf{A}_i^p = \mathbf{F}_{i, \mathbf{Y}}^p$, so that $\nabla \mathbf{F}_i^v \mathbf{d} = \mathbf{A}_i^v \nabla \mathbf{Y} \mathbf{d}$ and $\nabla \mathbf{F}_i^p \mathbf{d} = \mathbf{A}_i^p \nabla \mathbf{Y} \mathbf{d}$, respectively. Then the vector form of (62a)–(62b) is expressed as

$$0 = (\mathbf{W}, \mathbf{A}_0 \partial_t \mathbf{Y})_{\tilde{\Omega}} - (\partial_{x_i} \mathbf{W}, \mathbf{A}_i \mathbf{Y})_{\tilde{\Omega}} - (\mathbf{W}, \mathbf{Z})_{\tilde{\Omega}} + \langle \mathbf{W}, \tilde{\mathbf{H}} \rangle_{\tilde{\Gamma}} + (\mathcal{L} \mathbf{W}, \tau \mathbf{A}_0^{-1} (\mathcal{L}_t \mathbf{Y} - \mathbf{Z}))_{\tilde{\Omega}}. \quad (67)$$

5.2 Shallow water equations

In the case of the shallow water equations, it is important to take into consideration the complications that nonlinearities involve. We seek to construct a generalized vector of

boundary conditions $\tilde{\mathbf{H}}$, which enforces the modified boundary conditions on the surrogate boundary $\tilde{\Gamma}$.

We will decompose this problem into a series of subproblems which will be instrumental in forming the appropriate $\tilde{\mathbf{H}}$ for each of the six types of boundary conditions considered here. Let us begin by boundary conditions which enforce the value $h = \eta_* - z$ of the water column height on some portion Γ_* of the boundary Γ , where $\Gamma_* = \Gamma_D$ or $\Gamma_* = \Gamma_{I;sup}$. Then, on $\tilde{\Gamma}_*$ we will have

$$\begin{aligned} \frac{1}{2}g h^2(\tilde{\mathbf{x}}) &\approx \frac{1}{2}g h^2(\mathbf{x}) - gh(\tilde{\mathbf{x}})\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d} \\ &\approx \frac{1}{2}g (\eta_*(\mathbf{x}) - z(\mathbf{x}))^2 - gh(\tilde{\mathbf{x}})\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}, \end{aligned} \quad (68)$$

so that $\tilde{\mathbf{H}}^h$ can take the form

$$\begin{aligned} \tilde{\mathbf{H}}_*^h(\tilde{\mathbf{x}}) &= \mathbf{H}_*^h(\mathbf{M}(\tilde{\mathbf{x}})) - gh(\tilde{\mathbf{x}})\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d} \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) \end{Bmatrix} \\ &= \left(\frac{1}{2}g h_*^2 - gh(\tilde{\mathbf{x}})\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d} \right) \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) \end{Bmatrix}. \end{aligned} \quad (69)$$

Turning now our attention to the term $\tilde{\mathbf{H}}_N^v$ on Γ_N , its design is rather straightforward using (59) and other results already obtained in the acoustics case. A term that requires instead special care is the term $\tilde{\mathbf{H}}_*^v$, on the surrogate of $\Gamma_* = \Gamma_{O;sub}$, where we impose $h \mathbf{v} \cdot \mathbf{n} = m_*$. We start from (59) and derive, neglecting quadratic and higher-order terms:

$$\begin{aligned} h(\tilde{\mathbf{x}}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) &= h(\tilde{\mathbf{x}}) ((\mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}) \mathbf{n} + (\mathbf{v}(\tilde{\mathbf{x}}) \cdot \boldsymbol{\tau}) \boldsymbol{\tau}) \cdot \tilde{\mathbf{n}} \\ &\approx h(\tilde{\mathbf{x}}) (\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) - \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\ &\approx (h(\tilde{\mathbf{x}}) \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\ &\approx (h(\mathbf{x}) \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\ &= (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\ &\approx (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) (\mathbf{v}(\tilde{\mathbf{x}}) + (\nabla \mathbf{v}) \mathbf{d}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\ &\approx (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}}, \end{aligned} \quad (70)$$

where we removed the superscript j from $\boldsymbol{\tau}^j$ since the shallow water equations are inherently two-dimensional. Hence,

$$\tilde{\mathbf{H}}_*^v(\tilde{\mathbf{x}}) = ((m_* - (\nabla h \cdot \mathbf{d}) \mathbf{v} \cdot \mathbf{n} - h \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}}) \begin{Bmatrix} 1 \\ \mathbf{v} \end{Bmatrix}, \quad (71)$$

where as usual, $\mathbf{n}(\tilde{\mathbf{x}}) = \mathbf{n}(\mathbf{M}(\tilde{\mathbf{x}}))$, and $m_*(\tilde{\mathbf{x}}) = m_*(\mathbf{M}(\tilde{\mathbf{x}}))$.

In the case of a subcritical inlet $\Gamma_* = \Gamma_{I;sub}$, one additional boundary condition $\mathbf{v} \cdot \boldsymbol{\tau} = 0$ on the tangential component of velocity needs to be imposed on the surrogate boundary. We start from (70) but we consider a revised approximation for $\mathbf{v} \cdot \boldsymbol{\tau} = 0$, in

the term

$$\begin{aligned}
h(\tilde{\mathbf{x}}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) &\approx (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\
&\approx (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (\mathbf{v}(\mathbf{x}) \cdot \boldsymbol{\tau}(\mathbf{x}) - (\boldsymbol{\tau})^T \nabla \mathbf{v} \mathbf{d}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\
&\approx (m_* - (\nabla h(\tilde{\mathbf{x}}) \cdot \mathbf{d}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \mathbf{n}(\mathbf{x}) - h(\tilde{\mathbf{x}}) \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h(\tilde{\mathbf{x}}) (-\boldsymbol{\tau})^T \nabla \mathbf{v} \mathbf{d}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \\
&= \tilde{m}_*, \tag{72}
\end{aligned}$$

$$\begin{aligned}
h(\tilde{\mathbf{x}}) \mathbf{v}(\tilde{\mathbf{x}}) \cdot \tilde{\mathbf{n}}(\tilde{\mathbf{x}}) \mathbf{v}(\tilde{\mathbf{x}}) &\approx \tilde{m}_* (\mathbf{v}(\mathbf{x}) - \nabla \mathbf{v} \mathbf{d}) \\
&\approx \tilde{m}_* ((\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}) \mathbf{n} + (\mathbf{v}(\mathbf{x}) \cdot \boldsymbol{\tau}) \boldsymbol{\tau} - \nabla \mathbf{v} \mathbf{d}) \\
&\approx \tilde{m}_* ((\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}) \mathbf{n} - \nabla \mathbf{v} \mathbf{d}) \\
&\approx \tilde{m}_* (((\mathbf{v}(\tilde{\mathbf{x}}) + \nabla \mathbf{v} \mathbf{d}) \cdot \mathbf{n}) \mathbf{n} - \nabla \mathbf{v} \mathbf{d}) . \tag{73}
\end{aligned}$$

Then we obtain the following revised term $\tilde{\mathbf{H}}_*^v$:

$$\tilde{\mathbf{H}}_*^v(\tilde{\mathbf{x}}) = ((m_* - (\nabla h \cdot \mathbf{d}) \mathbf{v} \cdot \mathbf{n} - h \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h (-\boldsymbol{\tau})^T \nabla \mathbf{v} \mathbf{d}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}} \left\{ \begin{array}{c} 1 \\ ((\mathbf{v}(\tilde{\mathbf{x}}) + \nabla \mathbf{v} \mathbf{d}) \cdot \mathbf{n}) \mathbf{n} - \nabla \mathbf{v} \mathbf{d} \end{array} \right\}, \tag{74}$$

where quadratic and higher-order terms can be neglected in the products.

It remains to consider the case of $\tilde{\mathbf{H}}_{I;sup}^v$ on $\Gamma_{I;sup}$, which can simply be handled using Taylor expansions. Namely,

$$\tilde{\mathbf{H}}_{I;sup}^v(\tilde{\mathbf{x}}) = ((\eta_{I;sup} - z) - \nabla h \cdot \mathbf{d}) (\mathbf{v}_{I;sup} - (\nabla \mathbf{v}) \mathbf{d}) \left\{ \begin{array}{c} 1 \\ \mathbf{v}_{I;sup} - (\nabla \mathbf{v}) \mathbf{d} \end{array} \right\}. \tag{75}$$

In conclusion, we obtain the following final variational statement

$$0 = (\mathbf{W}, \partial_t \mathbf{U} - \mathbf{Z})_{\tilde{\Omega}} - (\partial_{x_i} \mathbf{W}, \mathbf{F}_i)_{\tilde{\Omega}} + \langle \mathbf{W}, \tilde{\mathbf{H}} \rangle_{\tilde{\Gamma}} + (\mathcal{L} \mathbf{W}, \tau \hat{\mathbf{A}}_0^{-1} (\mathcal{L}_t \mathbf{Y} - \mathbf{Z}))_{\tilde{\Omega}} + (\partial_{x_i} \mathbf{W}, \nu g^{ij} \hat{\mathbf{A}}_0 \partial_{x_j} \mathbf{Y})_{\tilde{\Omega}}, \tag{76a}$$

with $\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^v + \tilde{\mathbf{H}}^h$ and

$$\tilde{\mathbf{H}}^v(\mathbf{Y}) = \left\{ \begin{array}{ll} ((\mathbf{v}_N - \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}}) \left\{ \begin{array}{c} h \\ h \mathbf{v} \end{array} \right\}, & \text{on } \Gamma_N, \\ \mathbf{v} \cdot \tilde{\mathbf{n}} \left\{ \begin{array}{c} h \\ h \mathbf{v} \end{array} \right\}, & \text{on } \Gamma_D, \\ \tilde{m}_{I;sub} \left\{ \begin{array}{c} 1 \\ ((\mathbf{v}(\tilde{\mathbf{x}}) + \nabla \mathbf{v} \mathbf{d}) \cdot \mathbf{n}) \mathbf{n} - \nabla \mathbf{v} \mathbf{d} \end{array} \right\}, & \text{on } \Gamma_{I;sub}, \\ (h_{I;sup} - \nabla h \cdot \mathbf{d}) (\mathbf{v}_{I;sup} - (\nabla \mathbf{v}) \mathbf{d}) \left\{ \begin{array}{c} 1 \\ \mathbf{v}_{I;sup} - (\nabla \mathbf{v}) \mathbf{d} \end{array} \right\}, & \text{on } \Gamma_{I;sup}, \\ \tilde{m}_{O;sub} \left\{ \begin{array}{c} 1 \\ \mathbf{v} \end{array} \right\}, & \text{on } \Gamma_{O;sub}, \\ \mathbf{v} \cdot \tilde{\mathbf{n}} \left\{ \begin{array}{c} h \\ h \mathbf{v} \end{array} \right\}, & \text{on } \Gamma_{O;sup} \end{array} \right., \tag{76b}$$

and

$$\tilde{\mathbf{H}}^h(\mathbf{Y}) = \begin{cases} \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \left(\frac{1}{2}g(\eta_D - z)^2 - gh\nabla h \cdot \mathbf{d}\right) \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_D, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \left(\frac{1}{2}g(\eta_{I;sup} - z)^2 - gh\nabla h \cdot \mathbf{d}\right) \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \frac{1}{2}gh^2 \begin{Bmatrix} 0 \\ \tilde{\mathbf{n}} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}, \end{cases} \quad (76c)$$

where

$$\tilde{m}_{I;sub} = (m_{I;sub} - (\nabla h \cdot \mathbf{d}) \mathbf{v} \cdot \mathbf{n} - h \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h (-\boldsymbol{\tau})^T \nabla \mathbf{v} \mathbf{d} \boldsymbol{\tau} \cdot \tilde{\mathbf{n}}, \quad (76d)$$

$$\tilde{m}_{O;sub} = (m_{O;sub} - (\nabla h \cdot \mathbf{d}) \mathbf{v} \cdot \mathbf{n} - h \mathbf{n}^T \nabla \mathbf{v} \mathbf{d}) \mathbf{n} \cdot \tilde{\mathbf{n}} + h (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\mathbf{n}}. \quad (76e)$$

5.3 Discrete conservation properties

Also in the case of the SB method statement of conservation can be derived for the variational formulations (67) and (76), by using the same test function \mathbf{W}_j defined in Section 4.3, but this time supported over $\tilde{\Omega}$ only. Namely, for (76), we have,

$$0 = (\mathbf{W}_j, \partial_t \mathbf{U} - \mathbf{Z})_{\tilde{\Omega}} + \langle \mathbf{W}_j, \tilde{\mathbf{H}} \rangle_{\tilde{\Gamma}}, \quad (77)$$

and a similar statement holds for (67).

5.4 C-property

The analysis of the C-property for the SB method is very similar to the one for the base conformal method, but with one important difference, in that now the boundary condition values are approximated with Taylor expansions, and for this reason the boundary terms $\mathbf{F}_i^h n_i$ and $\tilde{\mathbf{H}}$ in the expression

$$(\mathbf{W}, \partial_t \mathbf{U})_{\tilde{\Omega}} = (\mathbf{W}, \mathbf{Z}_o - \partial_{x_i} \mathbf{F}_i)_{\tilde{\Omega}} + \langle \mathbf{W}, \mathbf{F}_i^h n_i - \tilde{\mathbf{H}} \rangle_{\tilde{\Gamma}}, \quad (78)$$

analogous to (56), may not always simplify exactly. Comparing with (76b)–(76c), we have that perfect cancellations always occur on the boundaries Γ_N , $\Gamma_{I;sub}$, $\Gamma_{O;sub}$, and $\Gamma_{O;sup}$, which do not need further care. However, on the boundary $\tilde{\Gamma}_D$ and $\tilde{\Gamma}_{I;sup}$, the shifting of boundary conditions produces terms of the type $gh\nabla h \cdot \mathbf{d}$.

If the domain is defined in a way that $\nabla z = 0$ in correspondence of Γ_D and $\Gamma_{I;sup}$, then $\nabla h = \nabla(h + z) = \nabla \eta_0 = 0$, and we recover, in the end, $(\mathbf{W}, A_0 \partial_t \mathbf{Y})_{\tilde{\Omega}} = 0$. If however

this is not the case, and $\nabla z \neq 0$ on Γ_D or $\Gamma_{I;sup}$, then the SB method may not satisfy the C-property. For this reason, in general a corrected form of (68) should be used, namely:

$$\begin{aligned} \frac{1}{2}g h^2(\tilde{\mathbf{x}}) &= \frac{1}{2}g (\eta - z)^2(\tilde{\mathbf{x}}) \approx \frac{1}{2}g (\eta - \nabla\eta(\tilde{\mathbf{x}}) \cdot \mathbf{d} - z)^2(\mathbf{x}) \\ &= \frac{1}{2}g h_*^2(\mathbf{x}) - gh(\tilde{\mathbf{x}})\nabla\eta(\tilde{\mathbf{x}}) \cdot \mathbf{d} + O(\|\mathbf{d}\|^2), \end{aligned} \quad (79)$$

with the convention that $h_*(\mathbf{x}) = \eta_*(\mathbf{x}) - z(\mathbf{x})$. This modification allows to recover the C-property and is essential when the gradient of the topography/bathymetry is non-zero near boundaries, as in many applications.

6 Time integration

The integration in time of the equations is performed using the same algorithms presented in [50, 47], to which the reader can refer for more details, and the analysis of stability and convergence, for body-fitted conformal grids. These algorithms are space-time integrators that are implemented as explicit predictor/multi-corrector with mass lumping. Specifically, we use a second-order Petrov-Galerkin space-time method, which, if only one corrector pass is performed, exactly corresponds to a second-order Runge-Kutta integrator. Full details about its implementation for acoustic wave problems is found in [50]. The extension of this algorithm to the shallow water equations is straightforward.

7 Numerical results for the wave propagation problem

We present a number of numerical results, to confirm the proposed approach is a robust, stable, and accurate strategy for boundary condition enforcement on embedded boundaries. All computations were run with three corrector passes of the second-order Petrov-Galerkin space-time integrator (similar to RK2) proposed in [50] for a Courant-Lewy-Friedrichs (CFL) condition of 0.5. The density ρ and wave speed c_s are both equal to the unit constant. We pick the stability parameter $c_\tau = 0.3$ and mass matrices are lumped for all testes. We start with a battery of tests to verify the accuracy of the proposed method.

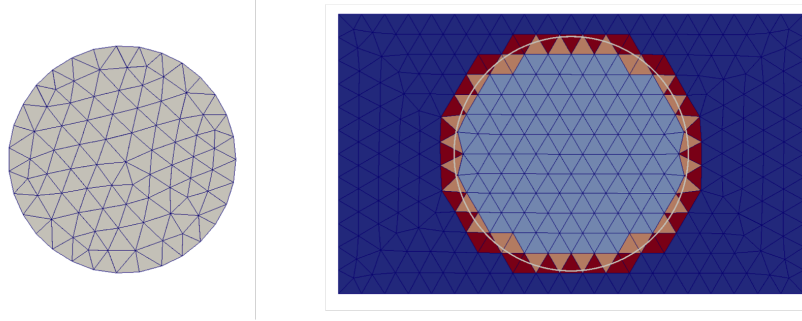


Figure 3: Linear acoustic wave convergence test. Computational domain geometry and grids at the coarsest level of refinement: conformal grid (left) and embedded grid (right). For the embedded grid, light blue indicates $\tilde{\Omega}$, that is the active elements inside the computational domain, dark blue indicates the elements outside, and red and orange indicate the elements intersecting the circular boundary (in white).

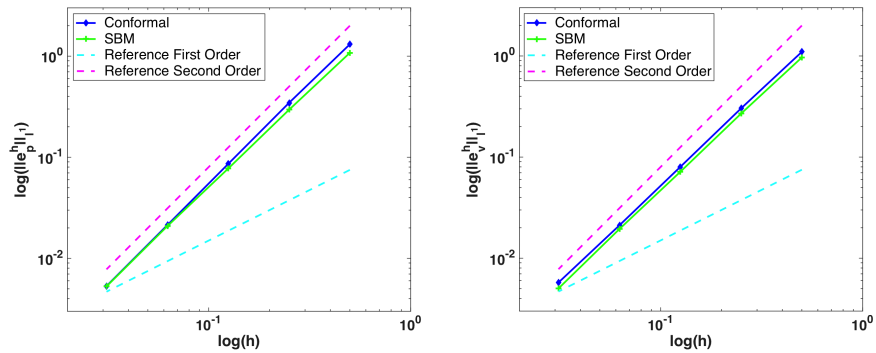


Figure 4: Convergence of the $l^1([0, T]:(L^2(\Omega)))$ -norm for the acoustic pulse propagation problem on an embedded disk domain with zero normal velocity boundary conditions. On the left, pressure, and on the right, velocity. A comparison between SB method and the conformal Nitsche method is shown for the errors of each solution field.

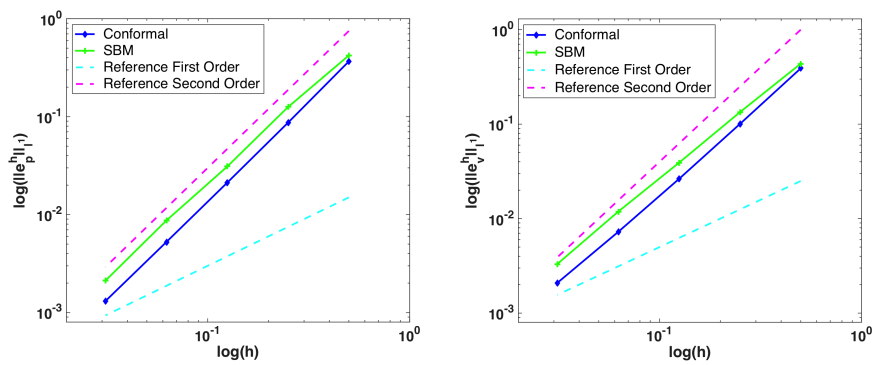


Figure 5: Convergence of the $l^1([0, T] : (L^2(\Omega)))$ -norm for the acoustic pulse propagation problem on an embedded disk domain with zero pressure boundary conditions. On the left, pressure, and on the right, velocity. A comparison between SB method and the conformal Nitsche method is shown for the errors of each solution field.

7.1 Convergence tests

7.1.1 Neumann boundary condition on an embedded disk

We consider a two-dimensional, radially symmetric, stationary acoustic wave propagation problem, where the computational domain is a disk of radius $R = 2.5$, as shown in Figure 3, with corresponding conformal and embedded grids. The exact solution is radially symmetric and given by

$$\begin{aligned} v_r(\mathbf{r}, t) &= \frac{1}{\rho c_s} J_1\left(\frac{z_{1,1}}{R} \mathbf{r}\right) \sin\left(\frac{c_s z_{1,1}}{R} t\right), \\ v_\theta(\mathbf{r}, t) &= 0, \\ p(\mathbf{r}, t) &= J_0\left(\frac{z_{1,1}}{R} \mathbf{r}\right) \cos\left(\frac{c_s z_{1,1}}{R} t\right), \end{aligned}$$

where v_r is the radial component of the velocity, v_θ is the tangential component of the velocity, and the scalar $z_{1,1} = 3.83170597020751$ is the first root of the first-kind Bessel function J_1 . Zero radial velocity boundary conditions are weakly enforced on the circle of radius R , using the SB method. Specifically, a background rectangular domain $[-5, 5] \times [-3, 3]$ is meshed using a fully unstructured triangular mesh, and its size is progressively and hierarchically refined from 0.5 to 0.0312 to check convergence rates. We consider the evolution of the problem until $T = 4$, that is, the instant of time at which the wave has reflected once against the boundary and is about to return to its original configuration.

Figure 4 shows the convergence of the $l^1([0, T] : (L^2(\Omega)))$ -norm of the pressure and velocity for this test. The rates of convergence are very close to the expected second order using lumped mass matrices. The slight degradation of the convergence rates may be a byproduct of a more pronounced accumulation of dispersion error over long time durations, a well-known issue in the case of mass lumping. For comparison, results computed on conformal (body-fitted) grids for the same problem are also presented in Figure 4, and are virtually indistinguishable from the results of the SB method.

7.1.2 Dirichlet boundary condition on an embedded disk

To test the SB method in an acoustic problem with Dirichlet boundary conditions, we consider the same domain geometry of the previous test, and we change the boundary conditions. Consequently, the exact solution has been altered to comply with a zero pressure boundary condition on the embedded disk or radius R ,

$$\begin{aligned} v_r(\mathbf{r}, t) &= \frac{1}{\rho c_s} J_1\left(\frac{z_{0,1}}{R} \mathbf{r}\right) \sin\left(\frac{c_s z_{0,1}}{R} t\right), \\ v_\theta(\mathbf{r}, t) &= 0, \\ p(\mathbf{r}, t) &= J_0\left(\frac{z_{0,1}}{R} \mathbf{r}\right) \cos\left(\frac{c_s z_{0,1}}{R} t\right), \end{aligned}$$

where $z_{0,1} = 2.40482555769577$ is the first root of the first-kind Bessel function J_0 . The solutions are computed until $T = 4$ with embedded and conformal grids.

We give the error of pressure and velocity in the $l^1([0, T] : (L^2(\Omega)))$ norm in Figure 5. Comparing the SB method with the reference Nitsche method on conformal grids, we observe that, although the errors in l^1 -norm associated with the SB are slightly larger, they still converge with order 1.8 and 1.7. We argue that this is actually a small price to

pay when weighted against the ease of implementation and simplicity of the method, if compared with strategies that require tedious integration along the true boundary of the computational domain.

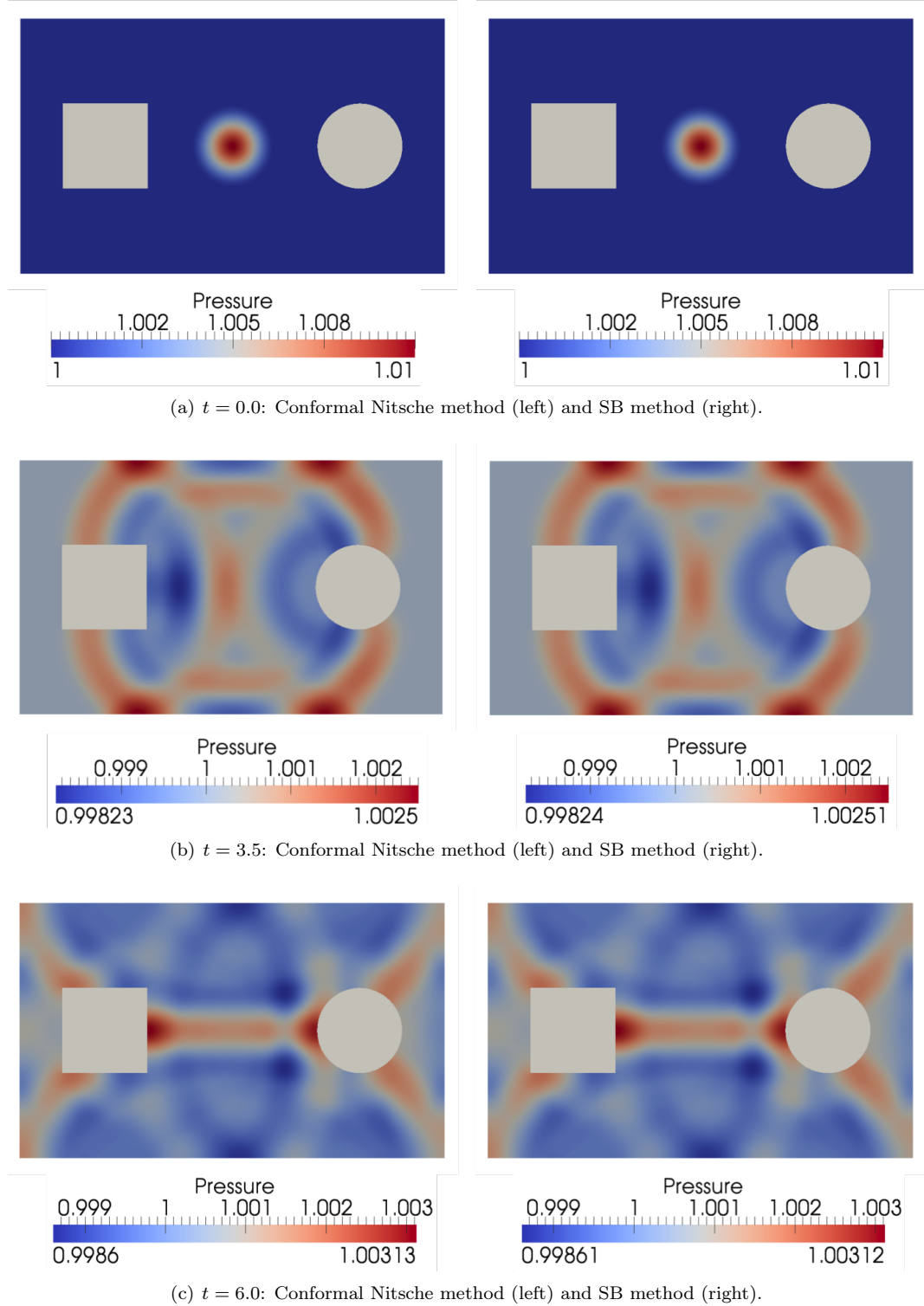
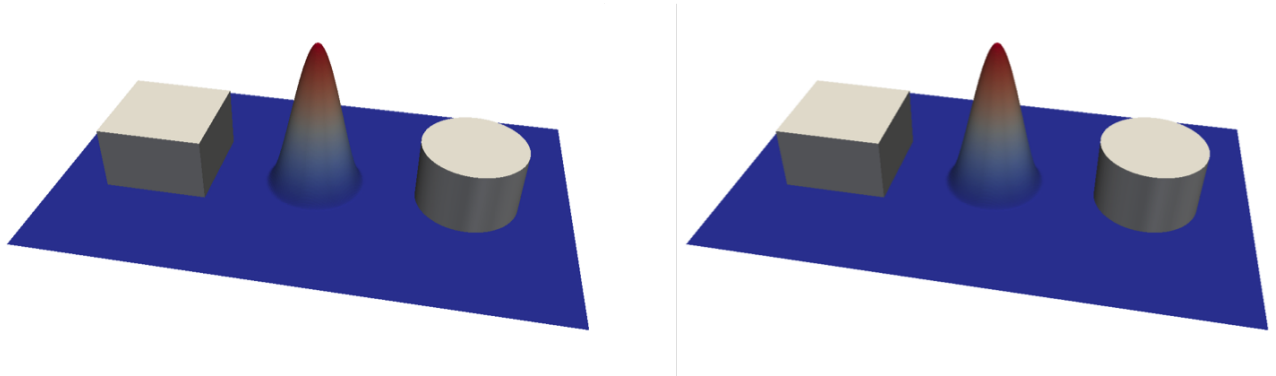
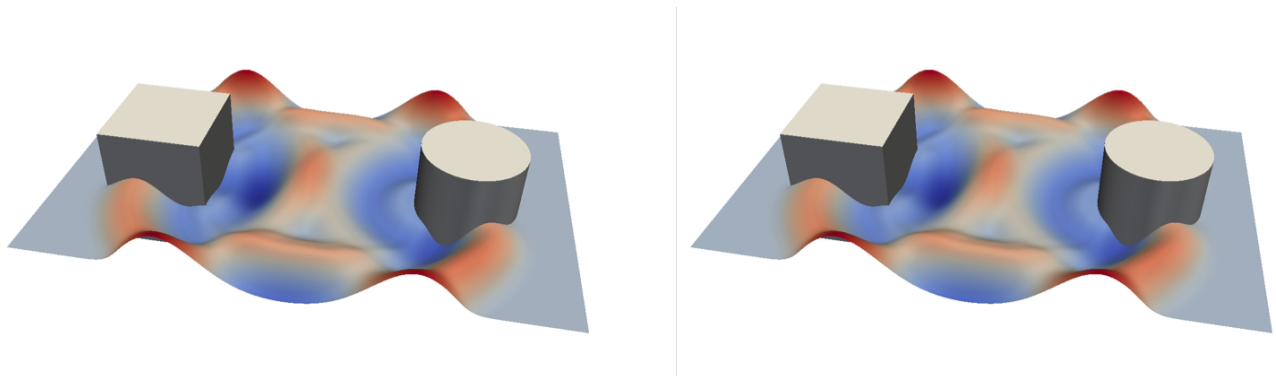


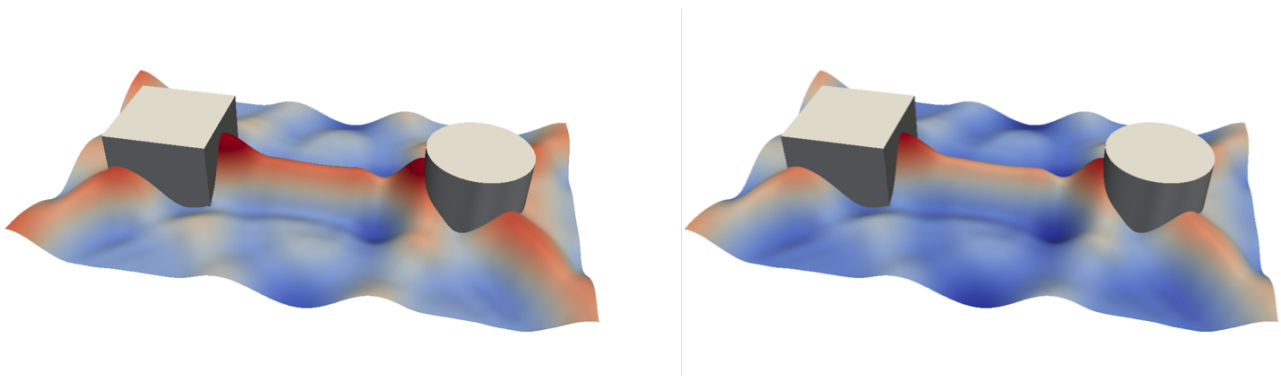
Figure 6: Comparison between pressure contours obtained with the conformal and SB methods, for the propagation of an acoustic wave in a complex two-dimensional domain at various times.



(a) $t = 0$: Conformal Nitsche method (left) and SB method (right).



(b) $t = 3.5$: Conformal Nitsche method (left) and SB method (right).



(c) $t = 6.0$: Conformal Nitsche method (left) and SB method (right).

Figure 7: Comparison between pressure elevation plots obtained with the conformal and SB methods, for the propagation of an acoustics wave in a complex two-dimensional domain at various times.

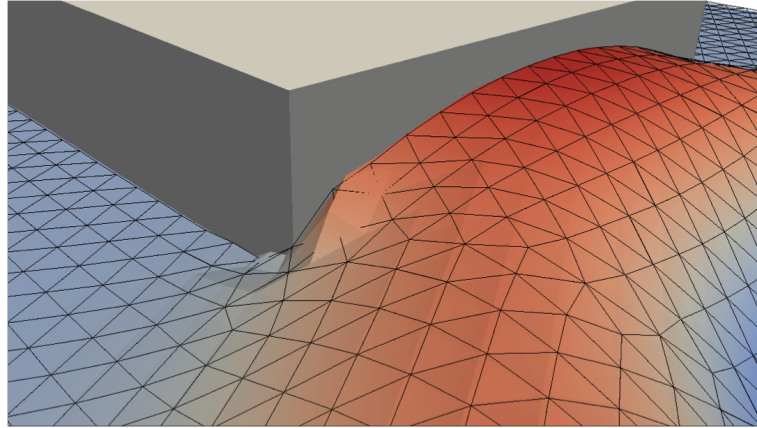
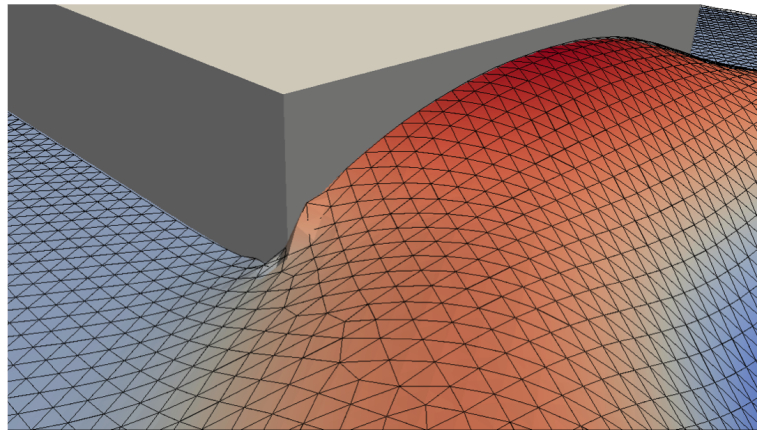
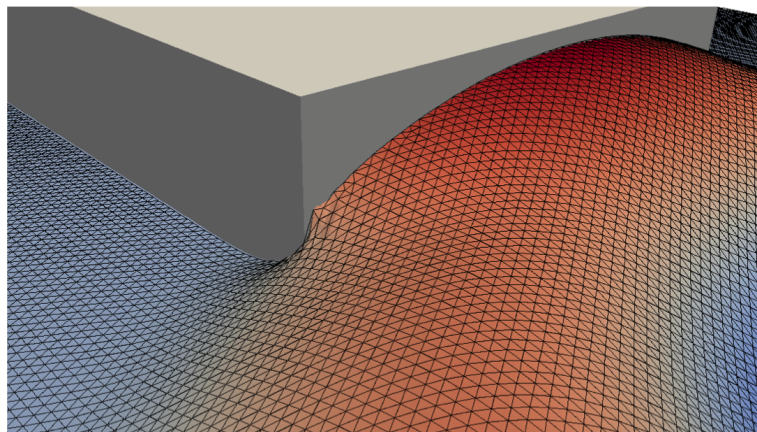
(a) mesh size $\Delta x = 0.125$.(b) mesh size $\Delta x = 0.0625$.(c) mesh size $\Delta x = 0.0312$.

Figure 8: Zoom of the elevation plot of the pressure solution near the rectangular boundary for various meshes. The results from SB method are shown as a solid contour, and are overlaid with the results of the conformal Nitsche method, shown as a black wireframe. The results of the conformal Nitsche and SB methods are difficult to tell apart.

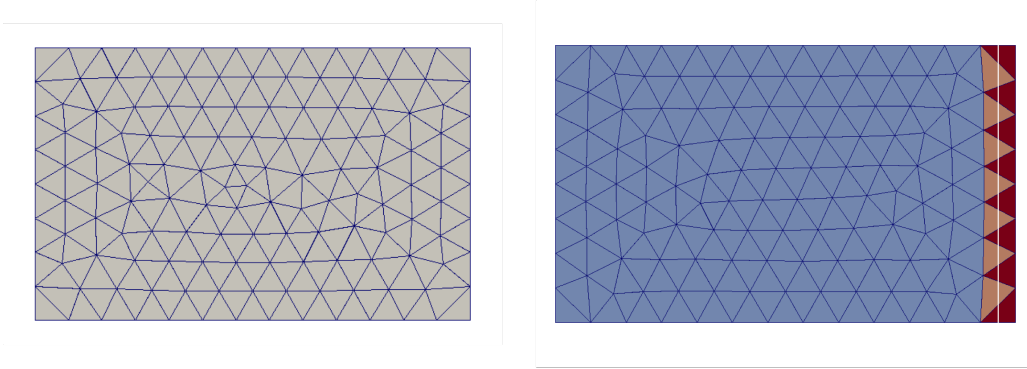


Figure 9: Nonlinear shallow water convergence test. Computational domain geometry and grids at the coarsest level of refinement: conformal grid (left) and embedded grid (right). For the embedded grid, light blue indicates $\tilde{\Omega}$, that is the active elements inside the computational domain, and red and orange indicate the (inactive) elements intersecting the circular boundary (in white).

7.2 Acoustic waves around circle/rectangle

This third test involves the interaction of radial acoustic wave with boundaries of more complex geometry. The problem domain consists of the rectangle $[-5, 5] \times [-3, 3]$, in which a circular hole is present, of radius 1.0 and center $(3, 0)$, as well as a square hole of side 2.0, and center $(-3, 0)$. The goal of this test is to validate the robustness of the proposed SB method in the presence of sharp corners, which may induce singularities in the velocity profile. The hump-like initial condition is given by the fields:

$$\mathbf{v} = \mathbf{0},$$

$$p = \begin{cases} 1 + A \left(1 + 2 \frac{r^3}{R^3} - 3 \frac{r^2}{R^2} \right), & r \leq 1, \\ 1, & \text{otherwise,} \end{cases}$$

where $A = 0.01$ is the amplitude, $R = 1.0$ is an intrinsic radius and $r = \sqrt{x^2 + y^2}$ is the magnitude of the radial vector. A zero normal velocity boundary condition is applied on all boundaries.

The computation using the SB method was performed on a grid that filled the entire rectangular domain $[-5, 5] \times [-3, 3]$ and of element size approximately 0.0312. In this test case, it is easy to observe that the solution is free from spurious oscillations throughout the many reverberations of the wave against the various boundaries, as shown in the pressure plots of Figures 6, 7 and 8.

The same problem is also solved on a conformal grid of almost identical mesh size that is attached to the geometrical shape of the problem domain. As shown in Figures 6, 7 and 8, the differences in pressure fields at various times are visually negligible.

An overlay of the results near the rectangular hole from the SB and conformal Nitsche methods is shown in Figure 8, where the solution of the SB method is shown as a solid surface and the solution of the conformal Nitsche method as a black wireframe. The two results show fairly good agreement.

8 Numerical results for shallow water flows

Also in the case of shallow water flows, we compare the performance of the proposed SB method against a reference Nitsche method on conformal grids of equivalent resolution. All computations were run with two corrector passes of the second-order Petrov-Galerkin space-time integrator proposed in [50] (very similar to RK2 explicit integrator) with a Courant-Lewy-Friedrichs (CFL) condition of 0.5. Moreover, we chose the stabilization parameter $c_\tau = 0.3$, and the primary variables $\mathbf{Y} = \{h, \mathbf{v}^T\}^T$ as solution variables. We also note that the discontinuity capturing operator is active only for the test of Sections 8.3 and 8.4.

8.1 C-property check

The C-property of the proposed SB method, discussed in Section 5.4, was validated in three numerical tests involving a body of water initially at rest in the rectangular domain $[-1, 1] \times [-1, 1]$. In the first test, the bathymetry of the river bed is given by a constant slope in the x -dimension, namely

$$z(x, y) = 1.0 - 0.25(x - x_{min}) . \quad (80)$$

In the second test, the river bed bathymetry has the shape of a hump, located at the center of the computational domain:

$$z(x, y) = \mathcal{I}_1^h [0.4 \exp(-10(x^2 + y^2))] , \quad (81)$$

where \mathcal{I}_1^h is the interpolation operator on the space of piecewise linear polynomials defined on the computational grid utilized in simulations. In these two tests, the boundaries are solid walls, and no friction is imposed. A third test is then considered, in which $\nabla z \neq 0$ and, in particular,

$$z(x, y) = 0.25 + 0.25 \mathcal{I}_1^h [\cos(\pi(x - x_{min}))] , \quad (82)$$

and fixed height boundary conditions are applied to the left and right hand side boundaries. Specifically, the correction suggested in (79) is used to maintain the C-property for the SB method. The other two sides are considered as solid walls.

For the SB method case, the right wall of the rectangular domain is embedded. The bathymetry of the river bed contributes to nonzero source terms S_{oi} . The water inside the domain is initially set at rest and the bathymetric bed is completely submerged. Ideally, the equilibrium state

$$h(x, y) + z(x, y) = 1.0 , \quad (83)$$

$$\mathbf{v} = \mathbf{0} , \quad (84)$$

should be maintained for all three bathymetries, as well as zero velocities everywhere. The simulations are carried out until $T = 100s$, using the SB method and the conformal Nitsche method, with a grid of size $h = 0.06$. The discontinuity capturing operator is switched off since there is no discontinuity forming. The L^2 -norm of height and velocity errors are presented in Table 1 for these three different bottom types. The water height and velocity are unperturbed and the zero velocity is reproduced within machine double precision in the L^2 -norm, for all tests and both numerical methods.

Table 1: C-property check. L^2 -norm of the error in height and velocity at time $T = 100s$, for the SB and the conformal Nitsche methods. The C-property is preserved within machine accuracy.

	<i>SB method</i>		<i>Conformal Nitsche method</i>	
	height	velocity	height	velocity
Constant slope bottom	1.60495e-12	6.2741e-13	6.29469e-14	6.90434e-13
Central hump bottom	4.04608e-12	7.78965e-13	9.81656e-14	9.18004e-13
Cosine shape bottom	6.20764e-14	7.17012e-13	5.56596e-14	7.38136e-13

8.2 Convergence tests for shallow water flows inside a rectangular domain

We present convergence tests for the SB method with three different types of boundary conditions: *impermeable wall*, *subcritical river inflow* and *subcritical river outflow*. The computation is confined inside the rectangular domain $[0, 8] \times [0, 5]$. In all simulations, and only for the case of the SB method, the right side of the rectangle - the set $\{8\} \times [0, 5]$ - is embedded, while for all other sides the boundary conditions are enforced weakly with a standard conformal Nitsche approach. Both the SB and the reference conformal Nitsche method were tested on five unstructured triangular grids of size $h = 0.625, 0.3125, 0.15625, 0.078125, 0.0390625$, respectively. We consider the evolution of the problem until $T = 3$. The computational domain and the coarsest level of refinement of the conformal and embedded grids are shown in Figure 9.

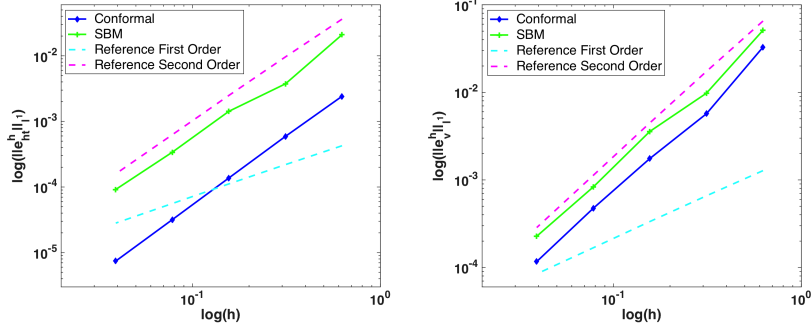


Figure 10: Convergence of the $l^1([0, T]; (L^2(\Omega)))$ -norm for the shallow water flows with zero normal velocity boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. Comparison between the SB and conformal Nitsche methods are presented.

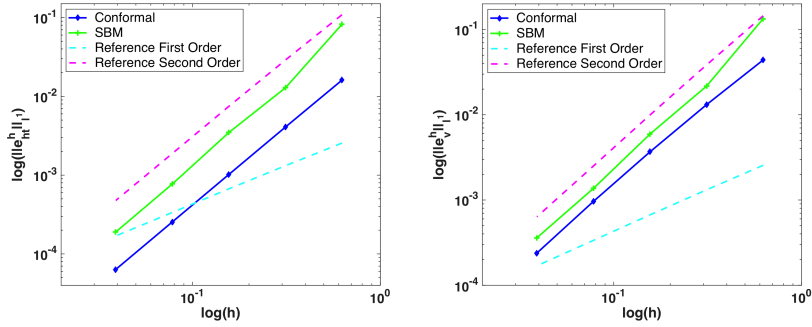


Figure 11: Convergence of the $l^1([0, T]; (L^2(\Omega)))$ -norm for the shallow water flows with an inflow boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. A comparison between the SB and the conformal Nitsche methods is shown.

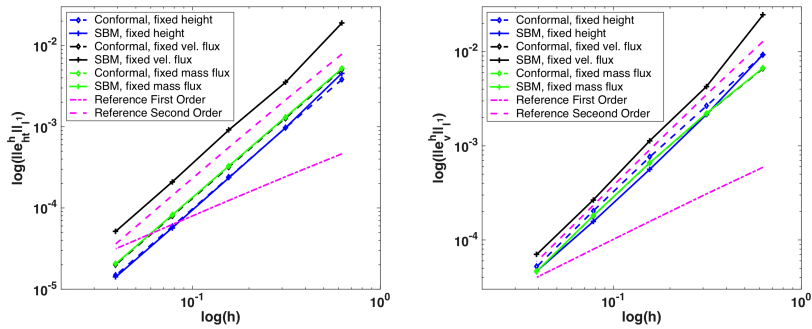


Figure 12: Convergence of the $l^1([0, T]; (L^2(\Omega)))$ -norm for the shallow water flows with outflow boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. Three different choices, namely fixed height, fixed velocity flux, and fixed mass flux, for outflow boundary conditions are tested and verified. A comparison between SB method and the conformal Nitsche method is shown.

8.2.1 Impermeable wall boundary conditions

In this case, zero normal velocity boundary conditions are applied to all boundaries of the rectangular domain. We consider the exact solution

$$h = 2 + 0.1 \sin(\omega t) \cos(\lambda x) , \quad (85a)$$

$$v_1 = 0.1 \sin(\lambda x) , \quad (85b)$$

$$v_2 = 0 , \quad (85c)$$

obtained with the method of manufactured solutions adding to the equations the source terms

$$S_1 = 0.1 (\cos(\lambda x) (2\lambda + \omega \cos(\omega t)) + 0.1\lambda \cos(2\lambda x) \sin(\omega t)) , \quad (86a)$$

$$S_2 = 0.1 \sin(\lambda x) (\lambda (0.005 - 2g + 0.015 \cos(2\lambda x)) \sin(\omega t) + 0.1 \cos(\lambda x) (4\lambda + \omega \cos(\omega t) - g\lambda \sin(\omega t) \sin(\omega t))) , \quad (86b)$$

$$S_3 = 0 , \quad (86c)$$

where we set, in particular, $\lambda = \pi/L$ and $\omega = 1$.

Figure 10 shows a comparison of the error convergence rates of the SB method and conformal Nitsche method. Both methods are nearly second order, but with differences in the L^2 -norm of the error that are more noticeable than in the acoustic case. These discrepancies are however within a reasonable range, and the accuracy sacrificed is a small price to pay for the simplicity of the proposed SB method.

8.2.2 River inflow boundary condition

In this case, zero normal velocity boundary conditions are applied to the top and bottom boundaries of the rectangular domain, and an outflow boundary condition on the mass flux $h \mathbf{v} \cdot \mathbf{n} = 0.2$ is applied to the left boundary. An inflow boundary condition ($h \mathbf{v} \cdot \mathbf{n} = -0.2$) is applied to the right wall, and implemented as embedded in the case of the SB method. The exact solution is given by

$$h = 2 - 0.1 \cos(\omega t) \sin(\lambda x) , \quad (87a)$$

$$v_1 = -0.1 \cos(\lambda x) , \quad (87b)$$

$$v_2 = 0 , \quad (87c)$$

obtained with the source terms:

$$S_1 = -0.1 (-0.1\lambda \cos(2\lambda x) \cos(\omega t) - \sin(\lambda x) (2\lambda + \omega \sin(\omega t))) , \quad (88a)$$

$$S_2 = -0.1 \cos(\lambda x) (\lambda (-0.005 + 2g + 0.015 \cos(2\lambda x)) \cos(\omega t) - 0.1g\lambda \cos(\omega)^2 \sin(\lambda x) + 0.1 \sin(\lambda x) (4\lambda + \omega \sin(\omega t))) , \quad (88b)$$

$$S_3 = 0 , \quad (88c)$$

where $\lambda = \pi/(2L)$ and $\omega = 1$. Note that the inflow boundary condition is enforced using the two separate conditions

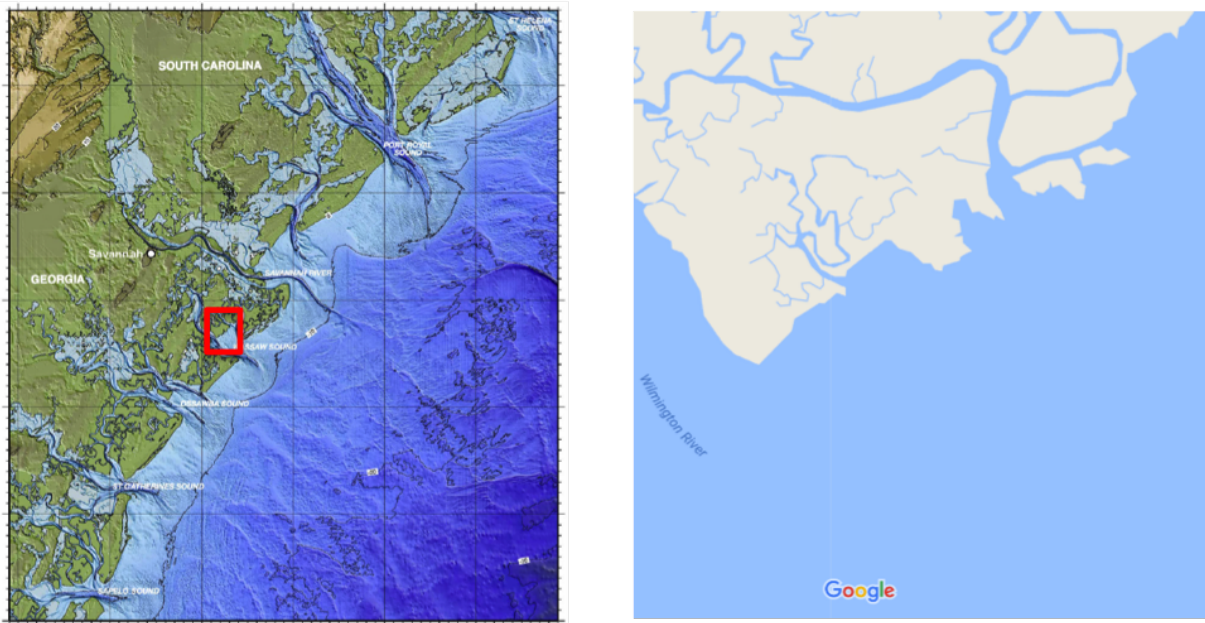
$$h \mathbf{v} \cdot \mathbf{n} = 0.2 \quad \text{and} \quad \mathbf{v} \cdot \boldsymbol{\tau} = 0 , \quad (89)$$

as suggested by the theory of Riemann invariants. As shown in Figure 11, the differences in errors between the conformal Nitsche method and the SB method are small, and both methods yield the expected second-order convergence rate.

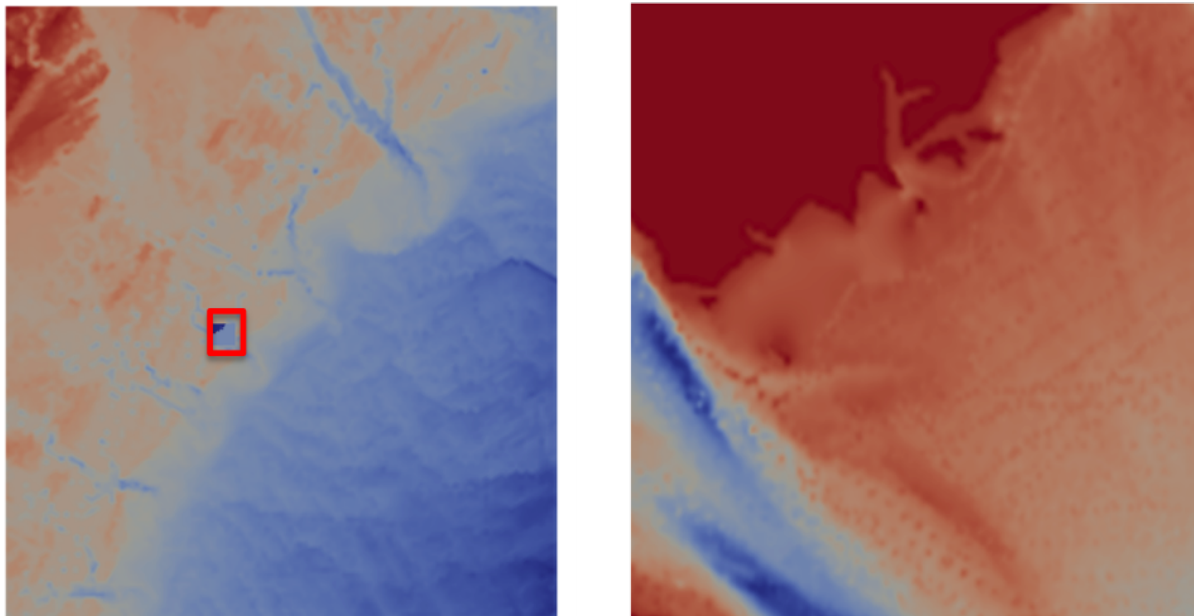
8.2.3 River outflow boundary condition

In this case, zero normal velocity boundary conditions are applied to the top and bottom boundaries of the rectangular domain. Outflow boundary conditions ($\mathbf{v} \cdot \mathbf{n} > 0$) are applied to the left and right boundaries of the rectangular domain. The exact solution is given by

$$\begin{aligned}h &= 2 - 0.1 \cos(\omega t) \sin(\lambda x), \\v_1 &= -0.1 \cos(\lambda x), \\v_2 &= 0,\end{aligned}$$



(a) Map view: Coastline of Savannah, GA (left) and zoom near the Cabbage island area (right), marked with a red box.



(b) DEM image: Coastline of Savannah, GA (left) and zoom near Cabbage island (right), marked with a red box. Zero elevation is used to represent the coastline

Figure 13: The coastline near Savannah (GA).

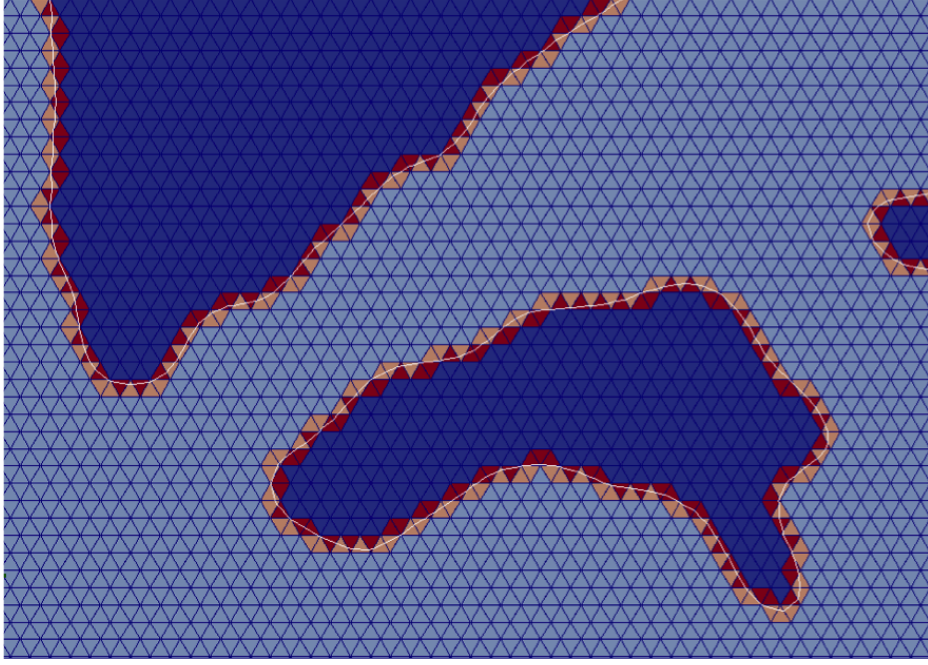


Figure 14: A closeup of the coastline and the embedded grids utilized in the computation. White solid lines represent the zero elevation isoline. Light blue indicates the active elements inside the computational domain, dark blue indicates the elements outside, and red and orange indicate the elements intersected by the coastline.

and is obtained with the source terms

$$S_1 = -0.1 (-0.1\lambda \cos(2\lambda x) \cos(\omega t) - \sin(\lambda x)(2\lambda + \omega \sin(\omega t))) , \quad (91a)$$

$$S_2 = -0.1 \cos(\lambda x) (\lambda (-0.005 + 2g + 0.015 \cos(2\lambda x)) \cos(\omega t) - 0.1g \lambda \cos(\omega)^2 \sin(\lambda x) + 0.1 \sin(\lambda x) (4\lambda + \omega \sin(\omega t))) , \quad (91b)$$

$$S_3 = 0 , \quad (91c)$$

where $\lambda = \pi/L$ and $\omega = 1$. The following three options for the imposition of the outflow boundary conditions are tested:

a.) *Mass flux:*

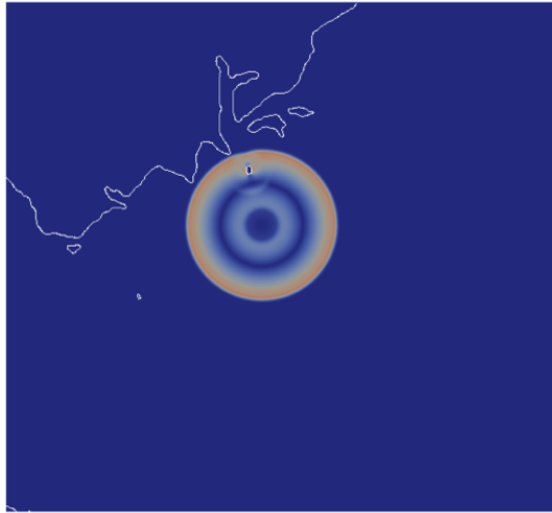
$$h \mathbf{v} \cdot \mathbf{n} = 0.2 ,$$

b.) *Fixed height:*

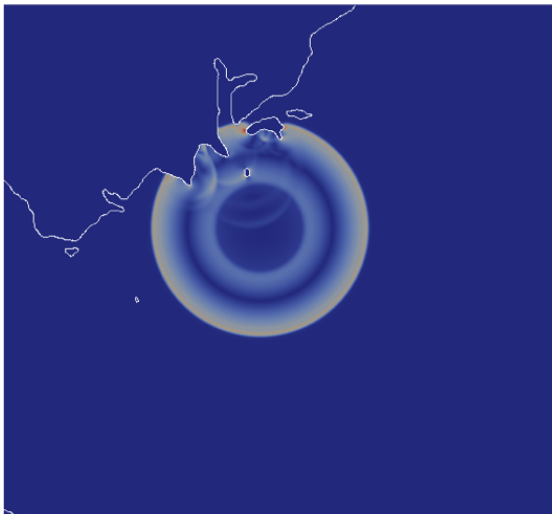
$$h = 2 ,$$

c.) *Velocity flux:*

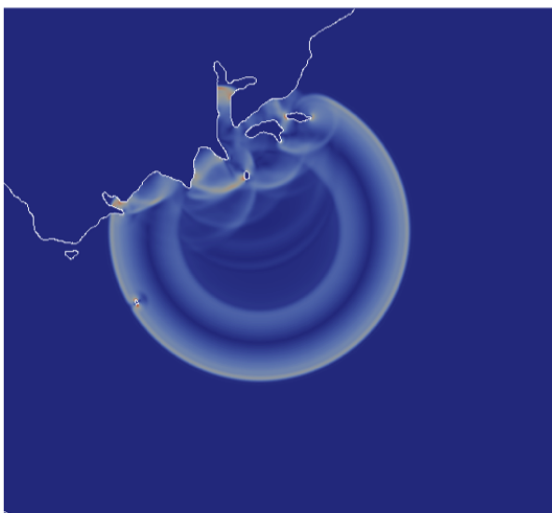
$$\mathbf{v} \cdot \mathbf{n} = 0.1 .$$

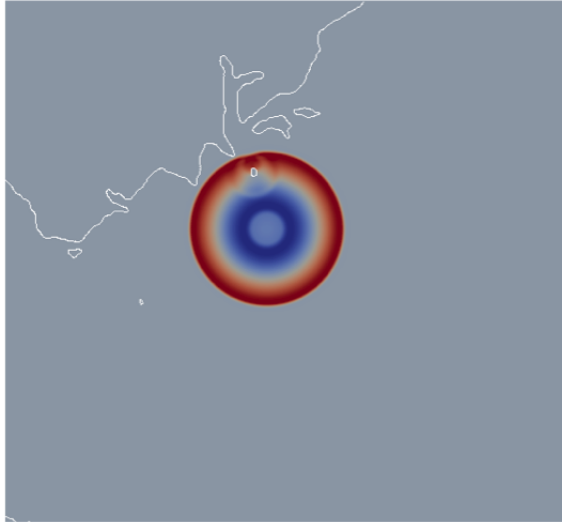
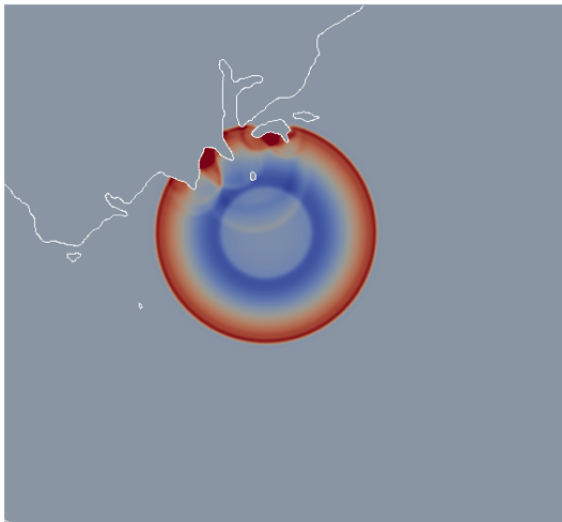
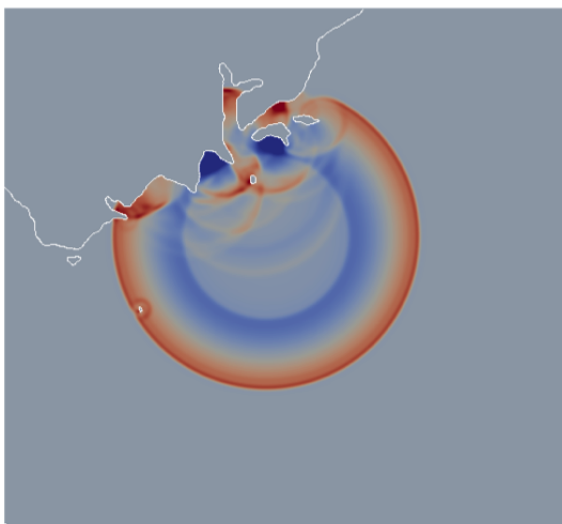


(a) $T = 120$



(b) $T = 190$



(a) $T = 120$ (b) $T = 190$ (c) $T = 280$

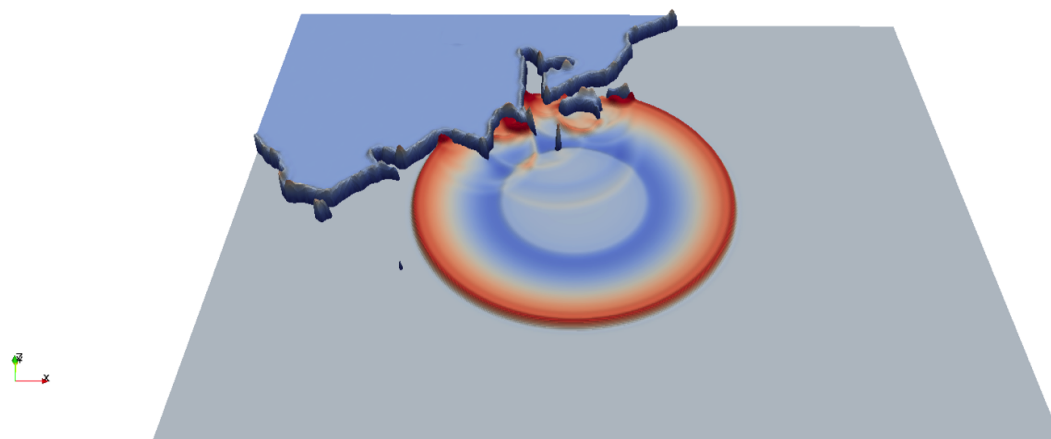
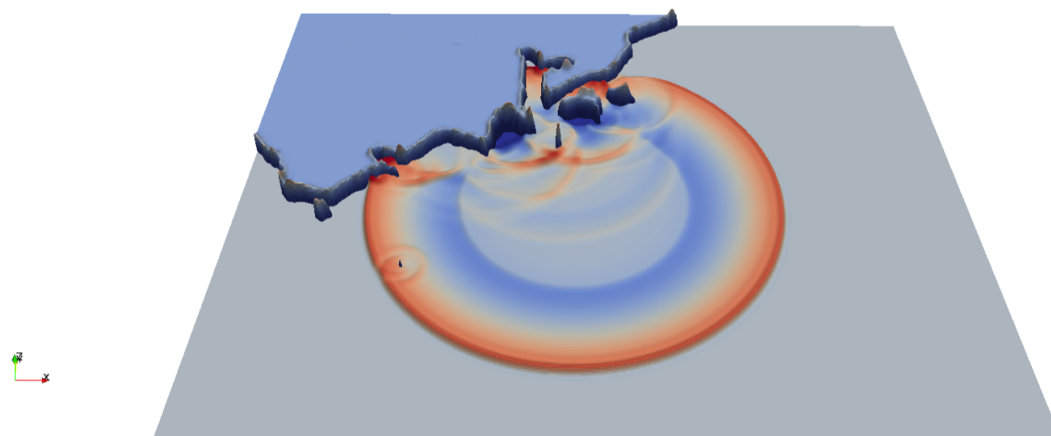
(a) $T = 220$ (b) $T = 300$

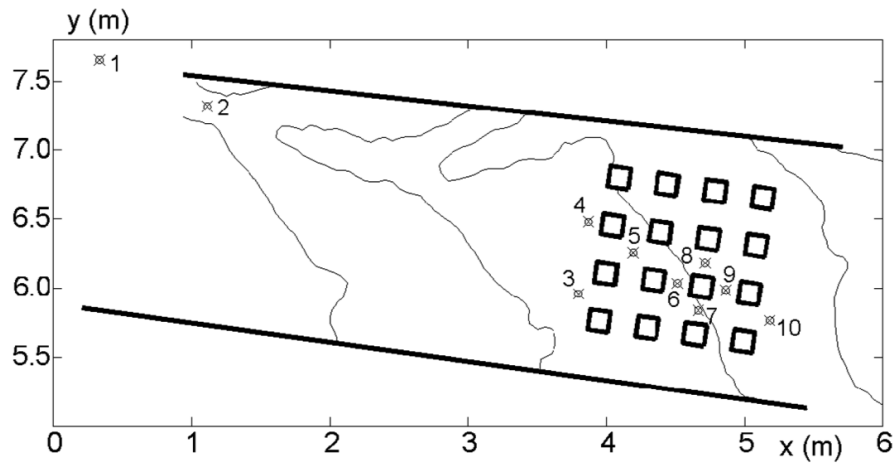
Figure 17: Height elevation plots for the propagation of a tsunami along the coastline near Savannah, (GA).

Results with all three previous options are illustrated in Figure 12. There is a discrepancy in the magnitude of errors between the Nitsche and conformal SB method in the case of the fixed height boundary conditions, while in all other cases the matching between the SB and conformal Nitsche method is very tight. In conclusion, we can say that the convergence rates are approximately second order for all implementations of the subcritical outflow boundary conditions.

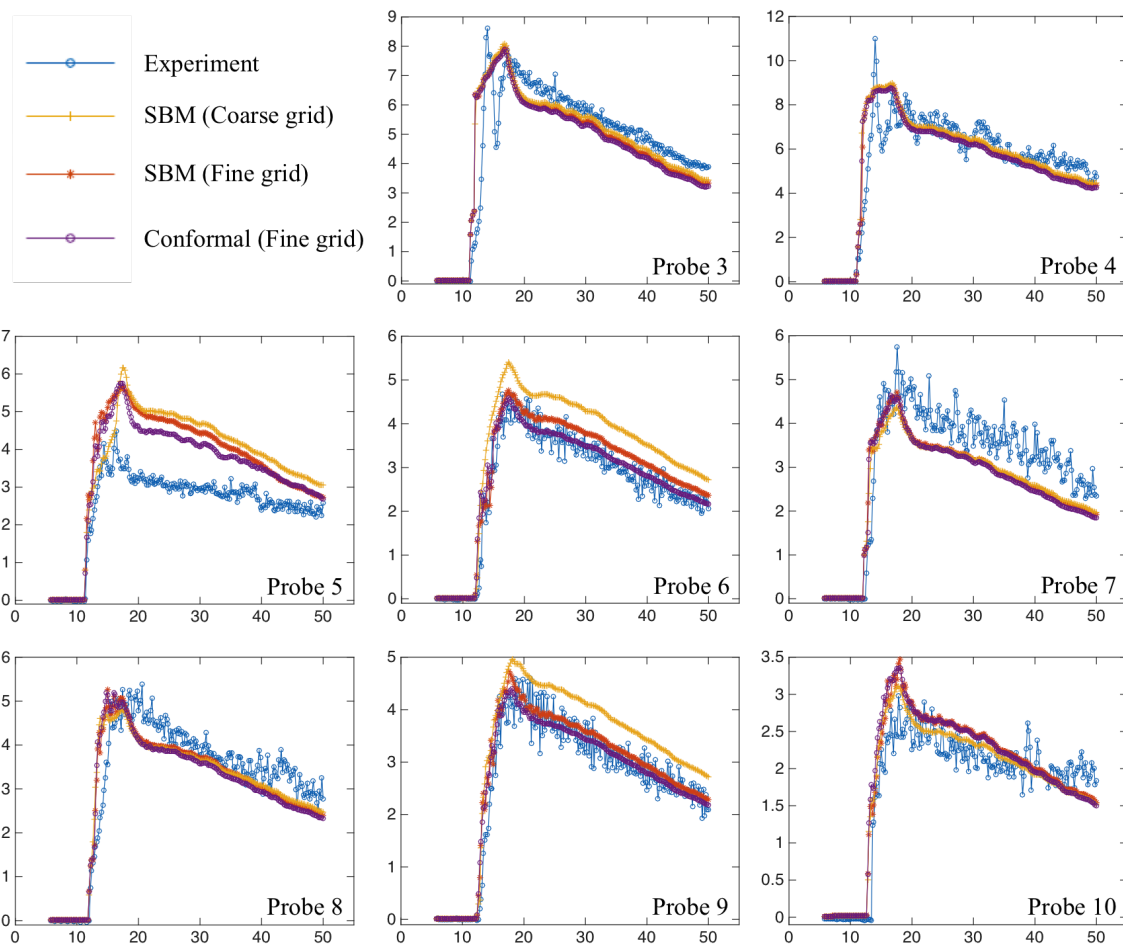
8.3 Coastline test

This test involves the propagation of a surface gravity-driven wave pulse against a portion of a coastline near Savannah, (GA). The coastline is reconstructed from the digital elevation map (DEM), as presented in Figure 13. The geometry of this problem is consid-

erably complicated and may require advanced meshing capability if a discretization using a conformal Nitsche method is pursued. This problem makes a good candidate for the proposed SB method. The spatial sampling interval of the DEM is about 8.75 meters in x-direction and 10.27 meters in y-direction. A triangulated grid of size 10 meters is used to fill the computation domain of $[35200, 39300] \times [-58400, -54600]$, with sufficient resolution. A closeup of the coastline and the embedded grid is presented in Figure 14.



(a) Modified valley geometry and building layout with locations of probes.



(b) Height of water at the probe locations 3 through 10, as depicted in Figure 18(a).

Figure 18: Toce valley [52] flash flood test. Comparison of water depth numerical predictions with experimental data for various probes. The horizontal axis represents time in seconds and the vertical axis represents water depth in centimeters.

The boundary conditions utilized are of the impermeable wall type for the coastline, and represent a simplification, since in general one should admit the possibility of flooding. However, when large areas of coastline that are not the primary focus of a simulation need to be modeled, this is often the boundary condition utilized. Flooding boundary/interface conditions will be considered in future work, due to their specific nature.

The initial solution utilized for this test is the same as the hump given in section 7.2, applied to the water height this time, with parameters $Amp = 1$, $R = 200$ and $r = \sqrt{(x - 37089)^2 + (y - 56250)^2}$.

The magnitude of the velocity and height are presented in Figure 15 and 16 at various times, while elevation plots of the height are presented in Figure 17. No oscillations are visible in this series of plots, in spite of the fact that the complex coastline chosen contains many corners. This test confirms that the proposed approach is a viable and robust strategy in the context of complex shallow water flows.

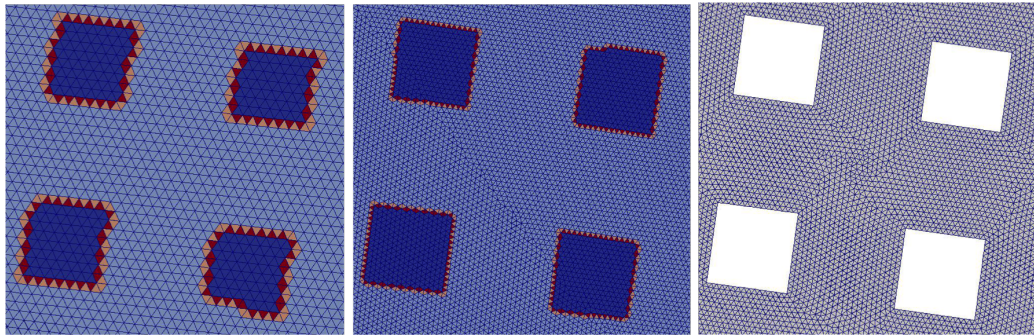


Figure 19: Toce valley flash flood test. Zoomed views of the embedded and conformal grids near the buildings in the urban area: (left) embedded grid (coarse), (middle) embedded grid (fine), (right) conformal grid. In the left and middle figures, light blue indicates the elements inside the computational domain, dark blue indicates the elements outside, and red and orange indicate the elements that intersect with the embedded boundaries.

8.4 Toce valley flash flood test: A simplified urban district layout

In [52], the authors conducted a series of flash flood flow experiments using a scaled model of the Toce river valley (Italy). The main goal of this work was the study of flow patterns in an urban environment, modeled by means of sixteen cubic concrete blocks of side $15cm$, arranged in a 4×4 grid. Two masonry walls were erected with the purpose of constraining the flow on either side of the urban area. The problem setup is shown in Figure 18. Ten probes measuring water heights are placed at various locations: in front of the first row of houses, in the middle of streets and in the wake of some of the buildings. The initial state of the model is dry, the pump discharge history data of "low flow" is used for the enforcement of upstream boundary condition, and supercritical outflow boundary conditions are imposed at the downstream end of the valley. The frictional coefficient is 0.0162 for the concrete bed. To resolve the possible numerical difficulties associated with the dry bed condition, for which the water level h is zero, a threshold value $\epsilon = 1.0e - 4$ is applied to h and entropy variables are used as suggested in [15].

In the computations presented here, three modifications and/or assumptions with

respect to the experimental setup are adopted to guarantee the well-posedness of numerical simulations. As shown in Figure 19, in the case of SBM, the buildings are embedded and two meshes of different size are used. A conformal mesh with size comparable to the finest SB method grid is used for comparison.

1. A constant inflow boundary condition is utilized at the inlet, since no specific information is available in the experiment.

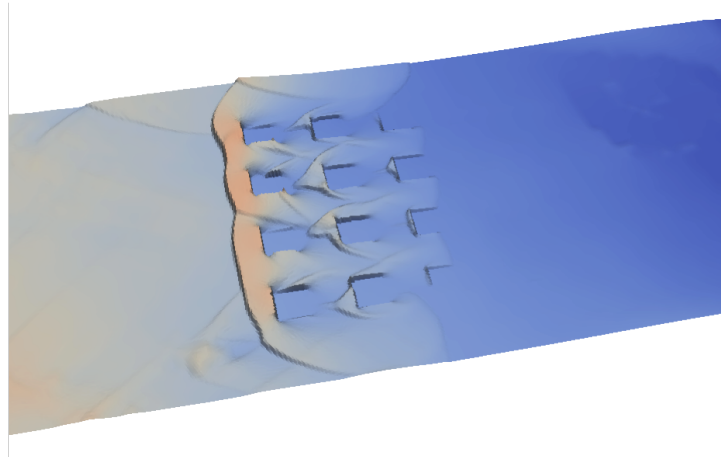
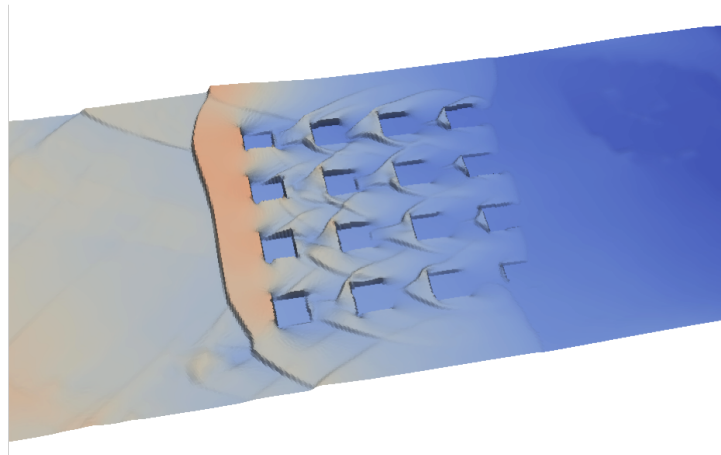
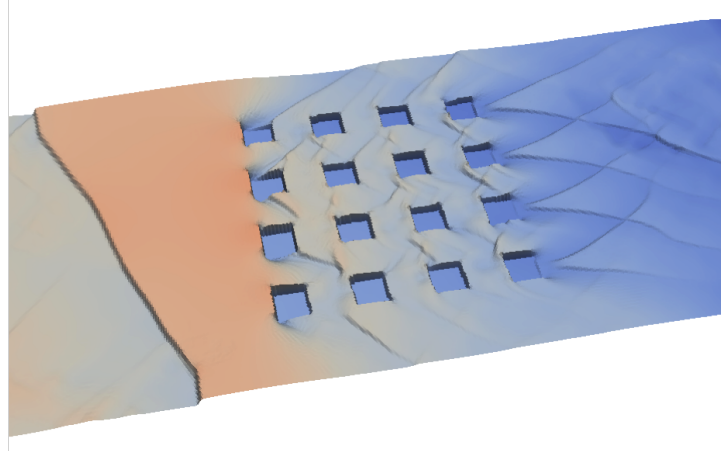
(a) $T = 6$ (b) $T = 7.5$ (c) $T = 9.8$

Figure 20: Toce valley [52] flash flood test. Free surface elevation plots for the SB method solution.

2. The location of the inlet was shifted to match the position of probe 2, since the bathymetry at the inlet does not seem to be represented accurately in the experimental data provided in [52]. Consequently, all the results from the numerical simulation reported here are offset by one second, which is the time taken by the water front to cover the distance between probe 1 and probe 2 in the experimental data. The initial flooding time is postponed to $t_0 = 4.6s$, since until that time, probe 2 shows particularly noisy discharge data.
3. The computational domain is extended to $x = 7.5$ with a smooth decrease in elevation, to ensure supercritical flow in the outlet area.

The free surface elevations near the urban district at various times are shown in Figure 20 for the SB method simulation. Figure 18 presents the water depths predicted by the SB method from 4.6 to 55 seconds, compared to the laboratory measurements and the Nitsche method results on conformal grids. The sampling time interval is 0.2 seconds. Overall, there is good agreement between the simulations of both the SB method and conformal Nitsche's method with the experiment, with the exception of probe 5, where the computational results match, but are markedly different from the experimental data. However, this fact is well documented in the literature of shallow water flow computations for this test [43, 30, 48], and the discrepancies observed in our computations are very similar to the ones observed in many other numerical simulations of this problem reported in the literature.

Acknowledgments

Guglielmo Scovazzi and the other members of his research team at Duke University (Ting Song and Alex Main) would like to gratefully acknowledge the support of the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research under Early Career Research Program Grant SC0012169, and the Army Research Office (ARO) under Grant W911NF-13-1-0452. Mario Ricchiuto has been partially funded by the TANDEM contract, reference ANR-11-RSNR-0023-01 of the French Programme Investissements d'Avenir. The support of the Associated Team HAMSTER between Duke University and the CARDAMOM INRIA team is also acknowledged. The authors would also like to thank Dr. Rodrigo Pedreros (at BRGM Orleans, France) and Dr. Chris Kees and Dr. Matthew Farthing (both at the Coastal and Hydraulics Laboratory, US Army ERDC, in Vicksburg, Mississippi) for valuable discussions.

References

- [1] Rémi Abgrall and Smadar Karni. Computations of compressible multifluids. *Journal of Computational Physics*, 169(2):594–623, 2001.
- [2] Chandrasekhar Annavarapu. *An Efficient Finite Element Method for Interface Problems*. PhD thesis, Duke University, 2013.
- [3] Chandrasekhar Annavarapu, Martin Hautefeuille, and John E Dolbow. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 225:44–54, 2012.
- [4] A. Bermudez and M.E. Vazquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049 – 1071, 1994.

-
- [5] Daniele Boffi and Lucia Gastaldi. A finite element approach for the immersed boundary method. *Computers & structures*, 81(8):491–501, 2003.
- [6] P. Brufau, P. Garcia-Navarro, and M.E. Vazquez-Cendon. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography. *Int. J. Numer. Meth. Fluids*, 45:1047–1082, 2004.
- [7] Erik Burman. Ghost penalty. *Comptes Rendus Mathematique*, 348(21-22):1217–1220, 2010.
- [8] Erik Burman and Peter Hansbo. Fictitious domain methods using cut elements: Iii. a stabilized nitsche method for stokes’ problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):859–874, 2014.
- [9] M.J. Castro, J. Gonzalez-Vida, and C. Pares. Numerical treatment of wet/dry fronts in shallow flows with a roe scheme. *Mathematical Models and Methods in Applied Sciences*, 16(6):897–931, 2006.
- [10] Derek M Causon, David M Ingram, and Clive G Mingham. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advances in water Resources*, 24(8):899–911, 2001.
- [11] Derek M Causon, David M Ingram, Clive G Mingham, G Yang, and Richard V Pearson. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in water resources*, 23(5):545–562, 2000.
- [12] Alexander Düster, Jamshid Parvzian, Zhengxiong Yang, and Ernst Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45):3768–3782, 2008.
- [13] Charbel Farhat, Arthur Rallu, and Sriram Shankaran. A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions. *Journal of Computational Physics*, 227(16):7674–7700, 2008.
- [14] Anita Hansbo and Peter Hansbo. An unfitted finite element method, based on nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47):5537–5552, 2002.
- [15] Guillermo Hauke. A symmetric formulation for computing transient shallow water flows. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):111–122, 1998.
- [16] Guillermo Hauke. A stabilized finite element method for the saint-venant equations with application to irrigation. *International journal for numerical methods in fluids*, 38(10):963–984, 2002.
- [17] Klaus Höllig. *Finite element methods with B-splines*. SIAM, 2003.
- [18] Klaus Höllig, Ulrich Reif, and Joachim Wipper. Weighted extended b-spline approximation of dirichlet problems. *SIAM Journal on Numerical Analysis*, 39(2):442–462, 2001.

-
- [19] Wang K. *A computational framework based on an embedded boundary method for nonlinear multi-phase fluid-structure interactions*. PhD thesis, Stanford University, 2011.
- [20] Y. Kagawa, T. Tsuchiya, T. Yamabuchi, H. Kawabe, and T. Fujii. Finite-element simulation of nonlinear sound-wave propagation. *Journal of Sound Vibration*, 154(1):125–145, 1992.
- [21] Seongwon Kang, Gianluca Iaccarino, Frank Ham, and Parviz Moin. Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method. *Journal of Computational Physics*, 228(18):6753–6772, 2009.
- [22] K. Kashiwama, H. Ito, M. Behr, and T. E. Tezduyar. 3-step explicit finite-element computations of shallow-water flows on a massively-parallel computer. *International Journal for Numerical Methods in Fluids*, 21(10):885–900, 1995.
- [23] K. Kashiwama and M. Kawahara. Selective lumping finite-element method for nearshore current. *International Journal for Numerical Methods in Fluids*, 4(1):71–97, 1984.
- [24] K. Kashiwama and M. Kawahara. Boundary-type finite-element method for surface-wave motion based on trigonometric function interpolation. *International Journal for Numerical Methods in Engineering*, 21(10), 1985.
- [25] K. Kashiwama and M. Kawahara. A boundary-type finite-element model for water-surface wave problems. *International Journal for Numerical Methods in Fluids*, 8(1):65–79, 1988.
- [26] K. Kashiwama, K. Saitoh, M. Behr, and T. E. Tezduyar. Parallel finite element methods for large-scale computation of storm surges and tidal flows. *International Journal for Numerical Methods in Fluids*, 24(12):1371–1389, 1997.
- [27] K. Kashiwama and M. Sakuraba. Adaptive boundary-type finite-element method for wave diffraction in harbors. *Computer Methods in Applied Mechanics and Engineering*, 112(1-4):185–197, 1994.
- [28] M. Kawahara, H. Sakurai, and K. Kashiwama. Boundary-type finite-element method for wave-propagation analysis. *International Journal for Numerical Methods in Fluids*, 8(5):559–578, 1988.
- [29] M. Kawahara, N. Takeuchi, and T. Yoshida. Two-step explicit finite-element method for tsunami wave-propagation analysis. *International Journal for Numerical Methods in Engineering*, 12(2):331–351, 1978.
- [30] B. Kim, B.F. Sanders, J.E. Schubert, and J.S. Famiglietti. Mesh type tradeoffs in 2d hydrodynamic modeling of flooding with a godunov-based flow solver. *Advances in Water Resources*, 68:42–61, 2014.
- [31] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. Accepted, *Journal of Computational Physics*, 2017.
- [32] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part II: Linear advection-diffusion and incompressible Navier-Stokes equations. Submitted, *Journal of Computational Physics*, 2017.

- [33] Hashem M Mourad, John Dolbow, and Isaac Harari. A bubble-stabilized finite element method for dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering*, 69(4):772–793, 2007.
- [34] J. A. Nitsche. Über ein Variationsprinzip zur Lösung Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ., Hamburg*, 36:9–15, 1971.
- [35] Jamshid Parvizian, Alexander Düster, and Ernst Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, 2007.
- [36] Charles S Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.
- [37] Fedkiw R., Aslam T., Merriman B., and Osher S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, 1999.
- [38] M. Ricchiuto. An explicit residual based approach for shallow water flows. *J.Comput.Phys.*, 80:306–344, 2015.
- [39] M. Ricchiuto and A. Bollerman. Stabilized residual distribution for shallow water simulations. *J.Comput.Phys.*, 228:1071–1111, 2009.
- [40] Mario Ricchiuto and Andreas Bollermann. Stabilized residual distribution for shallow water simulations. *Journal of Computational Physics*, 228(4):1071–1115, 2009.
- [41] T Rüberg and F Cirak. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209:266–283, 2012.
- [42] T Rüberg and F Cirak. A fixed-grid b-spline finite element technique for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 74(9):623–660, 2014.
- [43] B.F. Sanders, J.E. Schubert, and H.A. Gallegos. Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling. *Journal of Hydrology*, 362:19–38, 2008.
- [44] B Schott, U Rasthofer, V Gravemeier, and WA Wall. A face-oriented stabilized nitsche-type extended variational multiscale method for incompressible two-phase flow. *International Journal for Numerical Methods in Engineering*, 104(7):721–748, 2015.
- [45] B Schott and WA Wall. A new face-oriented stabilized xfm approach for 2d and 3d incompressible navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 276:233–265, 2014.
- [46] G. Scovazzi and B. Carnes. Weak boundary conditions for wave propagation problems in confined domains: Formulation and implementation using a variational multiscale method. *cmame*, 221-222:117–131, 2012.
- [47] G Scovazzi, T Song, and X Zeng. A velocity/stress mixed stabilized nodal finite element for elastodynamics: Analysis and computations with strongly and weakly enforced boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 325:532–576, 2017.

-
- [48] G. Vincent S.F. Sandra, L. Julien and Z. Yves. Two-dimensional shallow-water model with porosity for urban flood modelling. *Journal of Hydraulic Research*, 46:1:45–64, 2008.
- [49] W. D. Smith. Application of finite-element analysis to body wave-propagation problems. *Geophysical Journal of the Royal Astronomical Society*, 42(2):747–768, 1975.
- [50] T Song and G Scovazzi. A nitsche method for wave propagation problems in time domain. *Computer Methods in Applied Mechanics and Engineering*, 293:481–521, 2015.
- [51] S. Takase, K. Kashiya, S. Tanaka, and T. E. Tezduyar. Space-time SUPG formulation of the shallow-water equations. *International Journal for Numerical Methods in Fluids*, 64(10-12):1379–1394, 2010.
- [52] G. Testa, D. Zuccala, F. Alcrudo, J. Mulet, and S. Soares-Fraza. Flash flood flow experiment in a simplified urban district. *Journal of Hydraulic Research*, 45:37–44, 2007.
- [53] Kevin G Wang, Patrick Lea, Alex Main, Owen McGarity, and Charbel Farhat. Predictive simulation of underwater implosion: Coupling multi-material compressible fluids with cracking structures. In *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pages V08AT06A028–V08AT06A028. American Society of Mechanical Engineers, 2014.



**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200, Avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399