Contents lists available at ScienceDirect
Journal of Computational Physics
www.elsevier.com/locate/jcp

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

T. Song [a], A. Main [b], G. Scovazzi [a,b,*], M. Ricchiuto [c]

[a] *Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC 27708, United States of America*
[b] *Department of Civil & Environmental Engineering and Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC 27708, United States of America*
[c] *Team CARDAMOM, Inria Bordeaux - Sud-Ouest, 200 avenue de la vieille tour, 33405 Talence cedex, France*

ARTICLE INFO

Article history:
Received 22 December 2017
Received in revised form 25 April 2018
Accepted 27 April 2018
Available online xxxx

Keywords:
Embedded boundary conditions
Wave equation
Shallow water flows
Finite elements
Approximate boundary methods

ABSTRACT

We propose a new computational approach for embedded boundary simulations of hyperbolic systems and, in particular, the linear wave equations and the nonlinear shallow water equations. 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 body-fitted grids, which conform to the geometry of the computational domain boundaries, are also presented.

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, to linear acoustics and nonlinear shallow water flows.

Part of the motivation of this work is that time-domain acoustics in complex geometry still presents some computational challenges. Similar challenges are present for shallow water flows with 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 immersed/embedded methods, both from the mathematical and data structure perspectives.

* Corresponding author at: Department of Civil & Environmental Engineering, Duke University, Durham, NC 27708, United States of America.
E-mail address: guglielmo.scovazzi@duke.edu (G. Scovazzi).

https://doi.org/10.1016/j.jcp.2018.04.052
0021-9991/

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 [2,3,33], 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 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 [12,35], 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 [1,13], to compressible fluid-structure interaction in [19], and to multiphase fluid/structure interaction in [54]. 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 GFM 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 [10,11] 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 [20,46,47,50].

We implement the proposed approach for boundary conditions in the context of a stabilized finite element formulation, which takes inspiration specifically from [46,47,50] and more broadly from [15,16,22–29,40,49,52]. 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

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

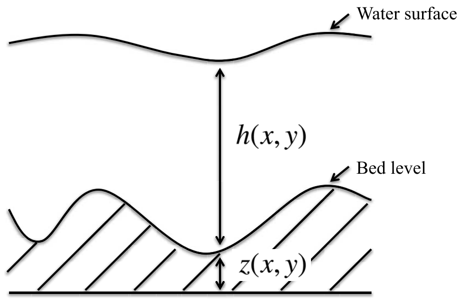


Fig. 1. Definition of water height h and the bathymetry z.

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. Additional problems in which the proposed method could be beneficial, left for future work, are human phonation simulations (fluid-structure-acoustics interaction) or dam break simulations (shallow water equations with complex topography).

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

$$\frac{\partial h}{\partial t} + \frac{\partial h v_1}{\partial x_1} + \frac{\partial h v_2}{\partial x_2} = 0\,, \tag{1a}$$

$$\frac{\partial}{\partial t}(h v_1) + \frac{\partial}{\partial x_1}\left(h v_1^2 + \frac{1}{2} g h^2\right) + \frac{\partial}{\partial x_2}(h v_1 v_2) = S_1\,, \tag{1b}$$

$$\frac{\partial}{\partial t}(h v_2) + \frac{\partial}{\partial x_1}(h v_1 v_2) + \frac{\partial}{\partial x_2}\left(h v_2^2 + \frac{1}{2} g h^2\right) = S_2\,. \tag{1c}$$

As illustrated in Fig. 1, h is the height of the water column, z is the bathymetry of the water bed, free surface level

$$\eta = h + z \tag{2}$$

and the two-dimensional position and velocity vectors are expressed in Cartesian coordinates as $\boldsymbol{x} = \{x_1\,,\, x_2\}$ and $\boldsymbol{v} = \{v_1\,,\, v_2\}$. $\partial/\partial t$ indicates derivation with respect to time, and

$$\boldsymbol{S} = \left\{ \begin{array}{c} S_1 \\ S_2 \end{array} \right\} = gh \left\{ \begin{array}{c} S_{o1} - S_{f1} \\ S_{o2} - S_{f2} \end{array} \right\}\,, \tag{3}$$

that is, $\boldsymbol{S}$ is the source vector containing the slope of the basin (e.g., river bed, ocean floor) in the ith 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 $\boldsymbol{Z}$ 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

$$\partial_t \boldsymbol{U} + \nabla \cdot \boldsymbol{G} = \boldsymbol{Z}\,, \tag{4a}$$

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

$$\boldsymbol{U} = \begin{Bmatrix} h \\ h\boldsymbol{v} \end{Bmatrix} , \quad \boldsymbol{G} = \boldsymbol{G}^{\boldsymbol{v}} + \boldsymbol{G}^h , \quad \boldsymbol{G}^{\boldsymbol{v}} = \boldsymbol{U} \otimes \boldsymbol{v} , \quad \boldsymbol{G}^h = \begin{Bmatrix} (\boldsymbol{0}_{2\times 1})^T \\ 1/2 g h^2 \boldsymbol{I}_{2\times 2} \end{Bmatrix} , \quad \text{and} \quad \boldsymbol{Z} = \begin{Bmatrix} 0 \\ \boldsymbol{S} \end{Bmatrix} . \tag{4b}$$

Here $\boldsymbol{U} \otimes \boldsymbol{v} = \boldsymbol{U}\boldsymbol{v}^T$ 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 $\boldsymbol{G}$, that is $\nabla \cdot \boldsymbol{G} = \partial_{x_j} G_{ij}$. The system of equations (4a) is also often expressed as

$$\partial_t \boldsymbol{U} + \partial_{x_i} \boldsymbol{F}_i = \boldsymbol{Z} , \tag{5}$$

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

$$\boldsymbol{F}_i = \boldsymbol{F}_i^{\boldsymbol{v}} + \boldsymbol{F}_i^h , \quad \boldsymbol{F}_i^{\boldsymbol{v}} = v_i \boldsymbol{U} , \quad \boldsymbol{F}_i^h = \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \boldsymbol{\delta}_i \end{Bmatrix} , \tag{6a}$$

with

$$\boldsymbol{\delta}_i = \begin{Bmatrix} \delta_{1i} \\ \delta_{2i} \end{Bmatrix} . \tag{6b}$$

Using any set of variables $\boldsymbol{X}$, it is possible to rewrite the vector form (5) in quasi-linear form as

$$\boldsymbol{A}_0 \, \partial_t \boldsymbol{X} + \boldsymbol{A}_i \, \partial_{x_i} \boldsymbol{X} = \boldsymbol{Z} . \tag{7}$$

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

$$\boldsymbol{A}_0 = \boldsymbol{I}_{3\times 3} , \tag{8a}$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ -v_1^2 + gh & 2v_1 & 0 \\ -v_1 v_2 & v_2 & v_1 \end{bmatrix} , \tag{8b}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 1 \\ -v_1 v_2 & v_2 & v_1 \\ -v_2^2 + gh & 0 & 2v_2 \end{bmatrix} . \tag{8c}$$

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

$$\hat{\boldsymbol{A}}_0 = \begin{bmatrix} 1 & 0 & 0 \\ v_1 & h & 0 \\ v_2 & 0 & h \end{bmatrix} , \tag{9a}$$

$$\hat{\boldsymbol{A}}_1 = \begin{bmatrix} v_1 & h & 0 \\ v_1^2 + gh & 2hv_1 & 0 \\ v_1 v_2 & hv_2 & hv_1 \end{bmatrix} , \tag{9b}$$

$$\hat{\boldsymbol{A}}_2 = \begin{bmatrix} v_2 & 0 & h \\ v_1 v_2 & hv_2 & hv_1 \\ v_2^2 + gh & 0 & 2hv_2 \end{bmatrix} . \tag{9c}$$

Adopting the definition of generalized entropy function in Tadmor [51],

$$\boldsymbol{E} = g\frac{h^2}{2} + h\frac{v_1^2 + v_2^2}{2}, \tag{10}$$

then the entropy variables $\boldsymbol{V}$ are given by

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{E}_{,U}^T \\ &= \begin{Bmatrix} gh - \frac{v_1^2 + v_2^2}{2} \\ v_1 \\ v_2 \end{Bmatrix} . \end{aligned} \tag{11}$$

It is also possible to rewrite the above system using the entropy variables $\boldsymbol{V}$ with

$$\tilde{\boldsymbol{A}}_0 = \frac{1}{g}\begin{bmatrix} 1 & v_1 & v_2 \\ & c^2+v_1^2 & v_1 v_2 \\ & \text{symm.} & c^2+v_2^2 \end{bmatrix},$$
$$\tilde{\boldsymbol{A}}_1 = \frac{1}{g}\begin{bmatrix} v_1 & c^2+v_1^2 & v_1 v_2 \\ & v_1(3c^2+v_1^2) & v_2(c^2+v_1^2) \\ & \text{symm.} & v_1(c^2+v_2^2) \end{bmatrix},$$
$$\tilde{\boldsymbol{A}}_2 = \frac{1}{g}\begin{bmatrix} v_2 & v_1 v_2 & c^2+v_2^2 \\ & v_2(c^2+v_1^2) & v_1(c^2+v_2^2) \\ & \text{symm.} & v_2(3c^2+v_2^2) \end{bmatrix}. \tag{12}$$

Noticing that matrix $\tilde{\boldsymbol{A}}_0$ is positive definite and $\tilde{\boldsymbol{A}}_i$ is symmetric for $i = 0, \dots, n_d$. Complete specification of the problem requires initial conditions on Ω, namely,

$$\boldsymbol{v}(t=0) = \boldsymbol{v}_0\,, \tag{13a}$$
$$h(t=0) = h_0\,, \tag{13b}$$

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. Here are a few typical examples:

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

$$\boldsymbol{v}\cdot\boldsymbol{n} = 0\,, \qquad \text{on } \Gamma_N\,, \tag{14}$$

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):

$$h = \eta_D - z\,, \qquad \text{on } \Gamma_D\,, \tag{15}$$

where the water height h is expressed as the difference of the free surface level η_D, defined according to (2), 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 ($|\boldsymbol{v}\cdot\boldsymbol{n}| \le c_s$):

$$\begin{cases} h\,\boldsymbol{v}\cdot\boldsymbol{n} = m_{I;sub}\,, \\ \quad\boldsymbol{v}\cdot\boldsymbol{\tau} = 0\,, \end{cases} \qquad \text{on } \Gamma_{I;sub}\,, \tag{16}$$

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 $|\boldsymbol{v}\cdot\boldsymbol{n}| \le c_s$. 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 $q = h\,\boldsymbol{v}\cdot\boldsymbol{n}$ in (16) can be replaced by the alternative conditions

$$\boldsymbol{v}\cdot\boldsymbol{n} = v_{I;sub}\,, \quad \text{or} \quad h = \eta_{I;sub} - z\,, \qquad \text{on } \Gamma_{I;sub}\,. \tag{17}$$

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

$$\begin{cases} \quad\boldsymbol{v} = \boldsymbol{v}_{I;sup}\,, \\ h = \eta_{I;sup} - z\,, \end{cases} \qquad \text{on } \Gamma_{I;sup}\,, \tag{18}$$

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 $|\boldsymbol{v}\cdot\boldsymbol{n}| > c_s$. 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 ($|\boldsymbol{v}\cdot\boldsymbol{n}| \le c_s$):

$$h\,\boldsymbol{v}\cdot\boldsymbol{n} = m_{O;sub}\,, \qquad \text{on } \Gamma_{O;sub}\,. \tag{19}$$

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:

$$\boldsymbol{v}\cdot\boldsymbol{n} = v_{O;sub}\,, \quad \text{or} \quad h = \eta_{O;sub} - z\,, \qquad \text{on } \Gamma_{O;sub}\,. \tag{20}$$

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.

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.

2.1. Linearized shallow waters equations: non-dissipative acoustic wave propagation

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

$$v_1 = \tilde{v}_1 \ll 1\,, \tag{21a}$$

$$v_2 = \tilde{v}_2 \ll 1\,, \tag{21b}$$

$$h = \bar{h} + \tilde{h}\,,\ \bar{h} = \text{const.}\,,\ \tilde{h} \ll 1. \tag{21c}$$

Substituting (21) in the shallow water equations (1a)–(1c) and neglecting higher-order terms, we obtain

$$\partial_t \tilde{h} + \nabla\cdot(\bar{h}\,\tilde{\boldsymbol{v}}) = 0\,, \tag{22a}$$

$$\partial_t(\bar{h}\,\tilde{\boldsymbol{v}}) + g\bar{h}\,\nabla\tilde{h} = \boldsymbol{S}\,. \tag{22b}$$

Removing, for the sake of simplicity, the tilde above $\boldsymbol{v}$, defining $p = g\tilde{h}$, dividing the first and second equation by $\bar{h}$, denoting by $c_s = \sqrt{\bar{h}g}$ the speed of waves, $\chi = c_s^{-2}$ and by $\boldsymbol{b} = \bar{h}^{-1}\,\boldsymbol{S}$, we have that (22) yield:

$$\chi\,\partial_t p + \nabla\cdot\boldsymbol{v} = 0\,, \tag{23a}$$

$$\partial_t \boldsymbol{v} + \nabla p = \boldsymbol{b}\,, \tag{23b}$$

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

$$\boldsymbol{v}(t=0) = \boldsymbol{v}_0\,, \tag{24a}$$

$$p(t=0) = p_0\,, \tag{24b}$$

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:

$$\boldsymbol{v}\cdot\boldsymbol{n}|_{\Gamma_N} = v_N\,(\boldsymbol{x},t)\ , \tag{25a}$$

$$p|_{\Gamma_D} = p_D\,(\boldsymbol{x},t)\ . \tag{25b}$$

Also (23) can be written in vector form. Denoting by $\boldsymbol{Y} = \{p, \boldsymbol{v}^T\}^T$ the solution state vector, we have

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

$$\partial_t \boldsymbol{U} + \partial_{x_i} \boldsymbol{F}_i = \boldsymbol{Z}\,, \quad \text{in}\,\Omega\,, \tag{26}$$

where

$$\boldsymbol{U} = \begin{Bmatrix} \chi\, p \\ \boldsymbol{v} \end{Bmatrix}, \qquad \boldsymbol{Z} = \begin{Bmatrix} 0 \\ \boldsymbol{b} \end{Bmatrix}, \tag{27}$$

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

$$\boldsymbol{F}_i = \boldsymbol{F}_i^{\boldsymbol{v}} + \boldsymbol{F}_i^{p}\,, \tag{28a}$$

$$\boldsymbol{F}_i^{\boldsymbol{v}} = v_i \begin{Bmatrix} 1 \\ \boldsymbol{0}_{n_d\times 1} \end{Bmatrix}, \tag{28b}$$

$$\boldsymbol{F}_i^{p} = p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \vdots \\ \delta_{n_d i} \end{Bmatrix}. \tag{28c}$$

Note the similarities in structure between (28) and (6). In order to recover the formalism of the quasi-linear shallow water equations found in equation (7), we can write

$$\boldsymbol{A}_0 \partial_t \boldsymbol{Y} + \boldsymbol{A}_i \partial_{x_i} \boldsymbol{Y} = \boldsymbol{Z}\,, \tag{29}$$

where

$$\boldsymbol{A}_0 = \begin{bmatrix} \chi & \boldsymbol{0}_{1\times n_d} \\ \boldsymbol{0}_{n_d\times 1} & \boldsymbol{I}_{n_d\times n_d} \end{bmatrix}, \tag{30a}$$

$$\begin{aligned} \boldsymbol{A}_i &= \boldsymbol{F}_{i,\boldsymbol{Y}} \\ &= \begin{bmatrix} 0 & \delta_{1i} & \cdots & \delta_{n_d i} \\ \delta_{1i} & & & \\ \vdots & & \boldsymbol{0}_{n_d\times n_d} & \\ \delta_{n_d i} & & & \end{bmatrix}. \end{aligned} \tag{30b}$$

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

3. Preliminaries: general notation, the true domain, the surrogate domain and maps

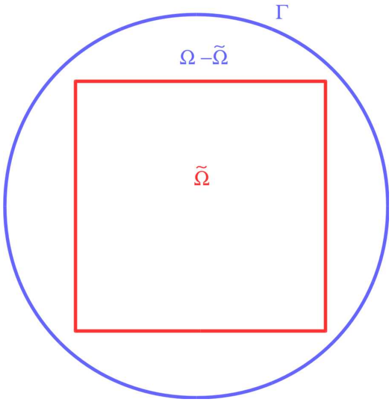
Consider the domain Ω, an open set in $\mathbb{R}^{n_d}$ with Lipschitz boundary $\Gamma = \partial\Omega$, where n_d is the number of space dimensions. Let $\mathscr{T}_h$ be a shape-regular triangulation (in the sense of Ciarlet) constituted by a family of non-overlapping n_d-partitions/elements Ω_e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

$$(v, w)_\omega = \int_\omega v\, w \qquad \text{and} \qquad (\boldsymbol{v}, \boldsymbol{w})_\omega = \int_\omega \boldsymbol{v}\cdot\boldsymbol{w} \tag{31a}$$

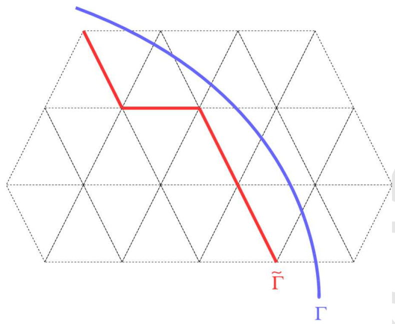
the $L^2(\omega)$- and $(L^2(\omega))^{n_d}$-inner products on the interior of ω and with

$$\langle v, w\rangle_\gamma = \int_\gamma v\, w \qquad \text{and} \qquad \langle \boldsymbol{v}, \boldsymbol{w}\rangle_\gamma = \int_\gamma \boldsymbol{v}\cdot\boldsymbol{w} \tag{31b}$$

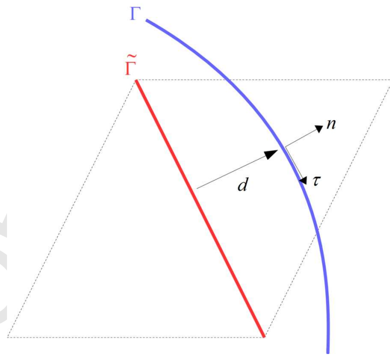
boundary functionals on γ. In particular, the space $L^2(\mathscr{E})$ refers to the set of functions whose traces are square integrable on the interior or exterior boundary set $\mathscr{E}$. Let $\|v\|^2 = (v, v)_\omega$, and let $W_j^k(\Omega)$ be the Sobolev space with norm



(a) The surrogate domain $\tilde{\Omega}$ and true domain Ω.



(b) The surrogate boundary $\tilde{\Gamma}$ and true boundary Γ.



(c) The distance vector $\boldsymbol{d}$, and the unit vectors normal and tangent to the true boundary Γ.

Fig. 2. The surrogate domain, the surrogate boundary, and the vector $\boldsymbol{d}$.

$$\|v\|_{W^k_j(\Omega)} = \left(\sum_{\alpha \le k} \|\boldsymbol{D}^\alpha_x v\|^j_{L^j(\Omega)} \right)^{1/j}, \quad 1 \le j < \infty . \tag{32}$$

For $j = 2$, let $H^k(\Omega) = W^k_2(\Omega)$ and set $\|v\|_{W^k_j} = \|v\|_k$.

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. The surrogate boundary $\tilde{\Gamma}$ is the boundary of a surrogate domain $\tilde{\Omega}$, composed of all the *full* elements contained in the true domain Ω. In other words, when all the cut elements of Ω are removed, we are left with $\tilde{\Omega}$. $\tilde{\Gamma}$ 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 closet 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, $\tilde{\boldsymbol{n}}$ indicates the unit outward-pointing normal to the surrogate boundary $\tilde{\Gamma}$, to be distinguished from the outward-pointing normal $\boldsymbol{n}$ of Γ. We now define the map

$$\boldsymbol{M} : \tilde{\Gamma} \to \Gamma , \tag{33a}$$

$$\tilde{\boldsymbol{x}} \mapsto \boldsymbol{x} , \tag{33b}$$

which maps a point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}$ on the surrogate boundary to a point $\boldsymbol{x} \in \Gamma$ on the true boundary. For example, the map $\boldsymbol{M}$ can also be built by means of the closest-point projection of points in $\tilde{\Gamma}$ onto Γ, as shown in Fig. 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 $\boldsymbol{M}$. The reason for the general smoothness of $\boldsymbol{M}$ is that the closest point projection is computed with respect to the true boundary, and yields a distance vector $\boldsymbol{d}$ in the direction of the normal $\boldsymbol{n}$ (rather than $\tilde{\boldsymbol{n}}$). The only situation when $\boldsymbol{M}$ loses its uniqueness is when a corner is present in Γ, but in this case, the definition of $\boldsymbol{M}$ can easily be adjusted by picking the vector $\boldsymbol{d}$ of minimal length among the possible choices at the corner. In practice, we observed that the implementation of $\boldsymbol{M}$ is robust even for the most complex geometries attempted.

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

$$\boldsymbol{d}_{\boldsymbol{M}}(\tilde{\boldsymbol{x}}) = \boldsymbol{x} - \tilde{\boldsymbol{x}} = [\,\boldsymbol{M} - \boldsymbol{I}\,](\tilde{\boldsymbol{x}}) . \tag{34}$$

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

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

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

$$\boldsymbol{n} \cdot \tilde{\boldsymbol{n}} \geq 0 \,, \tag{35}$$

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

$$\bar{\psi}(\tilde{\boldsymbol{x}}) \equiv \psi(\boldsymbol{M}(\tilde{\boldsymbol{x}})) \,. \tag{36}$$

For example, the unit normal vector $\boldsymbol{n}$ 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:

$$\bar{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) \equiv \boldsymbol{n}(\boldsymbol{M}(\tilde{\boldsymbol{x}})) \,, \tag{37a}$$

$$\bar{\boldsymbol{\tau}}_i(\tilde{\boldsymbol{x}}) \equiv \boldsymbol{\tau}_i(\boldsymbol{M}(\tilde{\boldsymbol{x}})) \,. \tag{37b}$$

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

$$\psi_{,\bar{n}}(\tilde{x}) = \nabla\psi(\tilde{x}) \cdot \bar{\boldsymbol{n}}(\tilde{x}) = \nabla\psi(\tilde{x}) \cdot \boldsymbol{n}(\boldsymbol{M}(\tilde{\boldsymbol{x}})) \,, \tag{38a}$$

$$\psi_{,\bar{\tau}_i}(\tilde{x}) = \nabla\psi(\tilde{x}) \cdot \bar{\boldsymbol{\tau}}_i(\tilde{x}) = \nabla\psi(\tilde{x}) \cdot \boldsymbol{\tau}_i(\boldsymbol{M}(\tilde{\boldsymbol{x}})) \,. \tag{38b}$$

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

$$\begin{aligned} u(\boldsymbol{x}) &= u(\tilde{\boldsymbol{x}}) + \nabla u(\tilde{\boldsymbol{x}}) \cdot (\boldsymbol{x} - \tilde{\boldsymbol{x}}) + O\left(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2\right) \\ &= u(\tilde{\boldsymbol{x}}) + \nabla u(\tilde{\boldsymbol{x}}) \cdot (\boldsymbol{M}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}) + O\left(\|\boldsymbol{M}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}\|^2\right) \\ &= u(\tilde{\boldsymbol{x}}) + \nabla u(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{d}(\tilde{\boldsymbol{x}}) + O\left(\|\boldsymbol{d}(\tilde{\boldsymbol{x}})\|^2\right) \,. \end{aligned} \tag{39}$$

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.

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

$$\mathscr{S}_p = \mathscr{V}_p = H^1(\Omega) \,, \tag{40a}$$

$$\mathscr{S}_{\boldsymbol{v}} = \mathscr{V}_{\boldsymbol{v}} = H^{div}(\Omega) \,. \tag{40b}$$

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 $\mathscr{S}_p^h \subset \mathscr{S}_p$, $\mathscr{V}_p^h \subset \mathscr{V}_p$, $\mathscr{S}_{\boldsymbol{v}} \subset \mathscr{S}_{\boldsymbol{v}}$ and $\mathscr{V}_{\boldsymbol{v}} \subset \mathscr{V}_{\boldsymbol{v}}$. Testing the strong form of the equation against the appropriate test functions and integrating by parts, we obtain:

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

$$0 = (\phi^h \,,\, \chi\, \partial_t p^h)_\Omega - (\nabla\phi^h \,,\, \boldsymbol{v}^h)_\Omega + \langle \phi^h \,,\, \widehat{\boldsymbol{v}\cdot\boldsymbol{n}} \rangle_\Gamma \,, \tag{41a}$$

$$0 = (\boldsymbol{\psi}^h \,,\, \partial_t \boldsymbol{v}^h - \boldsymbol{b})_\Omega - (\nabla \cdot \boldsymbol{\psi}^h \,,\, p^h)_\Omega + \langle \boldsymbol{\psi}^h \cdot \boldsymbol{n} \,,\, \hat{p} \rangle_\Gamma \,. \tag{41b}$$

The terms $\widehat{\boldsymbol{v}\cdot\boldsymbol{n}}$ 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:

$$\widehat{\boldsymbol{v}\cdot\boldsymbol{n}} = \begin{cases} \boldsymbol{v}\cdot\boldsymbol{n} & \text{on } \Gamma_D \,, \\ v_N & \text{on } \Gamma_N \,, \end{cases} \tag{42a}$$

$$\hat{p} = \begin{cases} p_D & \text{on } \Gamma_D \,, \\ p & \text{on } \Gamma_N \,. \end{cases} \tag{42b}$$

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

$$0 = (\phi\,,\ \chi\,\partial_t p)_\Omega - (\nabla\phi\,,\ \boldsymbol{v})_\Omega + \langle\phi\,,\ v_N\rangle_{\Gamma_N} + \langle\phi\,,\ \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\Gamma_D}\,, \tag{43a}$$

$$0 = (\boldsymbol{\psi}\,,\ \partial_t\boldsymbol{v} - \boldsymbol{b})_\Omega - (\nabla\cdot\boldsymbol{\psi}\,,\ p)_\Omega + \langle\boldsymbol{\psi}\cdot\boldsymbol{n}\,,\ p_D\rangle_{\Gamma_D} + \langle\boldsymbol{\psi}\cdot\boldsymbol{n}\,,\ p\rangle_{\Gamma_N}\,, \tag{43b}$$

which can be rewritten in system vector form as

$$0 = (\boldsymbol{W}\,,\ \boldsymbol{A}_0\partial_t\boldsymbol{Y} - \boldsymbol{Z})_\Omega - \left(\partial_{x_i}\boldsymbol{W}\,,\ \boldsymbol{F}_i\right)_\Omega + \langle\boldsymbol{W}\,,\ \widehat{\boldsymbol{F}_i n_i}\rangle_\Gamma, \tag{44a}$$

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

$$\boldsymbol{H}(\boldsymbol{Y}) = \widehat{\boldsymbol{F}_i n_i} = \begin{cases} v_N \begin{Bmatrix} 1 \\ \boldsymbol{0}_{n_d\times 1} \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}\,, & \text{on } \Gamma_N\,, \\[2ex] \boldsymbol{v}\cdot\boldsymbol{n} \begin{Bmatrix} 1 \\ \boldsymbol{0}_{n_d\times 1} \end{Bmatrix} + p_D \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}\,, & \text{on } \Gamma_D\,, \end{cases} \tag{44b}$$

which, recalling the identities

$$\boldsymbol{F}_i^p n_i = p \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}\,, \tag{45}$$

$$\boldsymbol{F}_i^{\boldsymbol{v}} n_i = \boldsymbol{v}\cdot\boldsymbol{n} \begin{Bmatrix} 1 \\ \boldsymbol{0}_{n_d\times 1} \end{Bmatrix}\,, \tag{46}$$

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

$$\boldsymbol{H}^{\boldsymbol{v}}(\boldsymbol{Y}) = \begin{cases} v_N \begin{Bmatrix} 1 \\ \boldsymbol{0}_{n_d\times 1} \end{Bmatrix}\,, & \text{on } \Gamma_N\,, \\[2ex] \boldsymbol{F}_i^{\boldsymbol{v}} n_i\,, & \text{on } \Gamma_D\,, \end{cases} \qquad \boldsymbol{H}^p(\boldsymbol{Y}) = \begin{cases} \boldsymbol{F}_i^p n_i\,, & \text{on } \Gamma_N\,, \\[2ex] p_D \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}\,, & \text{on } \Gamma_D\,, \end{cases} \tag{47a}$$

a form that highlights more clearly the type of boundary condition enforced, and the corresponding data. Then the boundary conditions (25a)–(25b) can be recast as

$$\text{Neumann condition on } \Gamma_N: \quad \boldsymbol{F}_i n_i = \boldsymbol{H} \ \Leftrightarrow\ \boldsymbol{v}\cdot\boldsymbol{n} = v_N\,, \tag{48a}$$

$$\text{Dirichlet condition on } \Gamma_D: \quad \boldsymbol{F}_i n_i = \boldsymbol{H} \ \Leftrightarrow\ p\boldsymbol{n} = p_D\boldsymbol{n}\,. \tag{48b}$$

Hence, it is possible to reformulate (43a)–(43b) using a stabilized formulation, analogous to the one pursued in [46,47,50] in the context of hyperbolic wave systems:

$$0 = (\boldsymbol{W}\,,\ \boldsymbol{A}_0\,\partial_t\boldsymbol{Y})_\Omega - \left(\partial_{x_i}\boldsymbol{W}\,,\ \boldsymbol{A}_i\boldsymbol{Y}\right)_\Omega - (\boldsymbol{W}\,,\ \boldsymbol{Z})_\Omega + \langle\boldsymbol{W}\,,\ \boldsymbol{H}\rangle_\Gamma + (\mathscr{L}\boldsymbol{W}\,,\ \tau\,\boldsymbol{A}_0^{-1}\,(\mathscr{L}_t\boldsymbol{Y} - \boldsymbol{Z}))_\Omega\,, \tag{49}$$

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

$$\begin{aligned} \mathscr{L}_t &= \boldsymbol{A}_0\partial_t + \mathscr{L}\,, \\ \mathscr{L} &= \boldsymbol{A}_i\partial_{x_i}\,. \end{aligned} \tag{50}$$

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 [46,47,50], and with close similarities to the variational forms developed in [15,16]:

$$0 = (\boldsymbol{W}\,,\, \partial_t \boldsymbol{U} - \boldsymbol{Z})_\Omega - \left(\partial_{x_i} \boldsymbol{W}\,,\, \boldsymbol{F}_i\right)_\Omega + \langle \boldsymbol{W}\,,\, \boldsymbol{H}\rangle_\Gamma + (\mathscr{L}^* \boldsymbol{W}\,,\, \tau\, \hat{\boldsymbol{A}}_0{}^{-1}\, (\mathscr{L}_t \boldsymbol{Y} - \boldsymbol{Z}))_\Omega + (\partial_{x_i} \boldsymbol{W}\,,\, \nu\, h_e^2 \hat{\boldsymbol{A}}_0\, \partial_{x_i} \boldsymbol{Y})_\Omega\,, \tag{51}$$

where $\mathscr{L}^* = \boldsymbol{A}_i^T \partial_{x_i}$, h_e is the characteristic mesh size and $\tau = c_\tau \frac{\Delta t}{2}$, again, is the stabilization characteristic time scale. The term $(\partial_{x_i} \boldsymbol{W}\,,\, \nu\, h_e^2 \hat{\boldsymbol{A}}_0\, \partial_{x_i} \boldsymbol{Y})_\Omega$ in (51) is a *discontinuity capturing operator*, with ν an artificial viscosity, defined as

$$\nu = c_\nu \max\left(0,\, \left[\frac{(\mathscr{L}_t \boldsymbol{Y} - \boldsymbol{Z}) \cdot \tilde{\boldsymbol{A}}_0^{-1}\, (\mathscr{L}_t \boldsymbol{Y} - \boldsymbol{Z})}{|\hat{\nabla}_\xi \boldsymbol{Y}|_{\hat{\boldsymbol{A}}_0}}\right]^{1/2} - \left|\frac{(\mathscr{L}_t \boldsymbol{Y} - \boldsymbol{Z}) \cdot \tau\, \tilde{\boldsymbol{A}}_0{}^{-1}\, (\mathscr{L}_t \boldsymbol{Y} - \boldsymbol{Z})}{|\hat{\nabla}_\xi \boldsymbol{Y}|_{\hat{\boldsymbol{A}}_0}}\right|\right). \tag{52}$$

Here, denoting $\xi_k\,, k = 1, 2$ the local element coordinates (in the elemental parent domain),

$$|\hat{\nabla}_\xi \boldsymbol{Y}|_{\hat{\boldsymbol{A}}_0} = \boldsymbol{V}_{,\boldsymbol{Y}} \boldsymbol{Y}_{,\xi_0} \cdot \hat{\boldsymbol{A}}_0 \boldsymbol{Y}_{,\xi_0} + g^{ij} \boldsymbol{V}_{,\boldsymbol{Y}} \boldsymbol{Y}_{,i} \cdot \hat{\boldsymbol{A}}_0 \boldsymbol{Y}_{,j}\,, \tag{53}$$

with $g^{ij} = [\xi_{k,i}\, \xi_{k,j}]^{-1}$ and

$$\boldsymbol{Y}_{,\xi_0} = \frac{\boldsymbol{Y}(t_{n+1}) - \boldsymbol{Y}(t_n)}{2}\,. \tag{54}$$

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

$$\boldsymbol{H}^{\boldsymbol{v}}(\boldsymbol{Y}) = \begin{cases} v_N \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_N\,, \\ \boldsymbol{v}\cdot\boldsymbol{n} \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_D\,, \\ m_{I;sub} \begin{Bmatrix} 1 \\ (\boldsymbol{v}\cdot\boldsymbol{n})\,\boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}\,, \\ \boldsymbol{v}_{I;sup}\cdot\boldsymbol{n} \begin{Bmatrix} \eta_{I;sup} - z \\ \left(\eta_{I;sup} - z\right)\, \boldsymbol{v}_{I;sup} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}\,, \\ m_{O;sub} \begin{Bmatrix} 1 \\ \boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}\,, \\ \boldsymbol{v}\cdot\boldsymbol{n} \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}\,, \end{cases} \qquad \boldsymbol{H}^h(\boldsymbol{Y}) = \begin{cases} \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_N\,, \\ \frac{1}{2} g\,(\eta_D - z)^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_D\,, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}\,, \\ \frac{1}{2} g \left(\eta_{I;sup} - z\right)^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}\,, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}\,, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}\,. \end{cases} \tag{55}$$

Observe the following equivalences of boundary condition expressions:

$$\underbrace{\boldsymbol{H}(\boldsymbol{Y})^{\boldsymbol{v}} + \boldsymbol{H}(\boldsymbol{Y})^{h}}_{\boldsymbol{H}(\boldsymbol{Y})} - \boldsymbol{F}_i n_i = \boldsymbol{0} \;\Leftrightarrow\; \begin{cases} (v_N - \boldsymbol{v}\cdot\boldsymbol{n}) \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix} = \boldsymbol{0} \Leftrightarrow \boldsymbol{v}\cdot\boldsymbol{n} = v_N\,, & \text{on } \Gamma_N\,, \\ \frac{1}{2} g\left((\eta_D - z)^2 - h^2\right) \begin{Bmatrix} 0 \\ \boldsymbol{n} \end{Bmatrix} = \boldsymbol{0} \Leftrightarrow h = \eta_D - z\,, & \text{on } \Gamma_D\,, \\ h\,\boldsymbol{v}\cdot\boldsymbol{n} = m_{I;sub} \quad \text{and} \quad \boldsymbol{v}\cdot\boldsymbol{\tau} = 0\,, & \text{on } \Gamma_{I;sub}\,, \\ \boldsymbol{v} = \boldsymbol{v}_{I;sup} \quad \text{and} \quad h = \eta_{I;sup} - z\,, & \text{on } \Gamma_{I;sup}\,, \\ \left(m_{O;sub} - h\,\boldsymbol{v}\cdot\boldsymbol{n}\right) \begin{Bmatrix} 1 \\ \boldsymbol{v} \end{Bmatrix} = \boldsymbol{0} \Leftrightarrow h\,\boldsymbol{v}\cdot\boldsymbol{n} = m_{O;sub}\,, & \text{on } \Gamma_{O;sub}\,, \\ \boldsymbol{0} = \boldsymbol{0} \Leftrightarrow \text{No conditions imposed}\,, & \text{on } \Gamma_{O;sup}\,. \end{cases} \tag{56}$$

To enhance stability, penalty terms are added to the variational form for the open sea and wall boundary conditions, respectively. Namely:

$$\langle \boldsymbol{\psi}\cdot\boldsymbol{n}\,,\, \alpha h\,(\boldsymbol{v}\cdot\boldsymbol{n} - v_N)\rangle_{\Gamma_N}\,, \quad \text{on } \Gamma_N\,, \tag{57a}$$

$$\langle \phi\,,\, \alpha\,(h - (\eta_D - z))\rangle_{\Gamma_D}\,, \quad \text{on } \Gamma_D\,. \tag{57b}$$

These penalty terms do not change the Euler–Lagrange equations associated to the variational formulation.

4.3. Discrete conservation properties

The variational formulation (51) naturally incorporates a conservation statement. Taking as test function $\boldsymbol{W} = \boldsymbol{1}_j$, where $\boldsymbol{1}_j$ is an array in which the j^{th} entry is unity and all the others are zero, we obtain:

$$0 = (\boldsymbol{1}_j\,,\, \partial_t \boldsymbol{U} - \boldsymbol{Z})_\Omega + \langle \boldsymbol{1}_j\,,\, \boldsymbol{H}\rangle_\Gamma\,, \tag{58}$$

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 $\boldsymbol{H}$ vanishes as well as the integral of the source vector $\boldsymbol{Z}$. A similar statement can be obtained for the acoustic system (49) (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

$$\boldsymbol{v} = 0\,, \tag{59a}$$

$$h + z = \eta_0\,(\text{constant})\,. \tag{59b}$$

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 (59), 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 (59) in (51) 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 $\boldsymbol{F}_i^{\boldsymbol{v}} = v_i\,\boldsymbol{U}$ 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 $\boldsymbol{Z}_o$ the simplified version of $\boldsymbol{Z}$ due to the previous simplifications. Now, if exact quadrature formulas are used for the evaluation of the terms $(\partial_{x_i}\boldsymbol{W}, \boldsymbol{F}_i^{\boldsymbol{h}})_\Omega$ and $(\boldsymbol{W}, \boldsymbol{Z}_o)_\Omega$, 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

$$(\boldsymbol{W}\,,\,\partial_t \boldsymbol{U})_\Omega = \left(\boldsymbol{W}\,,\,\boldsymbol{Z}_o - \partial_{x_i}\boldsymbol{F}_i\right)_\Omega + \langle \boldsymbol{W}\,,\,\boldsymbol{F}_i^h n_i - \boldsymbol{H}\rangle_\Gamma\,. \tag{60}$$

Let us focus our attention now on the term $\left(\boldsymbol{W}\,,\,\boldsymbol{Z}_o - \partial_{x_i}\boldsymbol{F}_i\right)_\Omega$. Note first that the first component of $\boldsymbol{Z}_o - \partial_{x_i}\boldsymbol{F}_i$, corresponding to the mass conservation equation, vanish, and that the if $\boldsymbol{\psi}$ 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 (59) holds. We are thus left with the boundary term $\langle \boldsymbol{W}\,,\,\boldsymbol{F}_i^h n_i - \boldsymbol{H}\rangle_\Gamma$, 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 (59). Hence, we conclude that the stabilized variational form detailed in (51) satisfies the C-property.

Remark 2. 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 [38,39] and consists in modifying the nodal values of the bathymetry as follows:

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

where on each element

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

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 $\boldsymbol{M}$ defined in (34) and the Taylor formula (39) 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 $\boldsymbol{d} = \boldsymbol{M}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}$, from $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_D$ to $\boldsymbol{x} = \boldsymbol{M}(\tilde{\boldsymbol{x}}) \in \Gamma_D$,

$$\begin{aligned} 0 &\approx p(\tilde{\boldsymbol{x}}) + \nabla p(\tilde{\boldsymbol{x}}) \cdot (\boldsymbol{x} - \tilde{\boldsymbol{x}}) - p_D(\boldsymbol{x}) + O\left(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2\right) \\ &= p(\tilde{\boldsymbol{x}}) + \nabla p(\tilde{\boldsymbol{x}}) \cdot (\boldsymbol{M}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}) - p_D(\boldsymbol{M}(\tilde{\boldsymbol{x}})) + O\left(\|\boldsymbol{M}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}\|^2\right) \\ &= p(\tilde{\boldsymbol{x}}) + \nabla p(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{d}(\tilde{\boldsymbol{x}}) - p_D(\boldsymbol{M}(\tilde{\boldsymbol{x}})) + O\left(\|\boldsymbol{d}(\tilde{\boldsymbol{x}})\|^2\right). \end{aligned} \tag{61}$$

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 $\bar{p}_D(\tilde{\boldsymbol{x}}) = p_D(\boldsymbol{M}(\tilde{\boldsymbol{x}}))$. Then the shifted boundary condition on surrogate Dirichlet-type boundary $\tilde{\Gamma}_D$ reads

$$p|_{\tilde{\Gamma}_D} = \bar{p}_D(\tilde{\boldsymbol{x}}) - \nabla p(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{d}(\tilde{\boldsymbol{x}}). \tag{62}$$

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 $\tilde{\boldsymbol{n}}$ to the surrogate boundary and the normal $\boldsymbol{n}$ to the true boundary do not coincide. This situation can be resolved by decomposing the unit normal vector $\tilde{\boldsymbol{n}}$ at $\tilde{\boldsymbol{x}}$ as $\tilde{\boldsymbol{n}} = (\tilde{\boldsymbol{n}} \cdot \boldsymbol{n})\boldsymbol{n} + (\tilde{\boldsymbol{n}} \cdot \boldsymbol{\tau}^j)\boldsymbol{\tau}^j$, where $\boldsymbol{n}$ is the normal to Γ_N and $\boldsymbol{\tau}^j$ $(j = 1, \cdots, n_d - 1)$ are the vectors tangent to Γ_N, respectively, and we also recall that $\boldsymbol{n}(\tilde{\boldsymbol{x}}) = \boldsymbol{n}(\boldsymbol{M}(\tilde{\boldsymbol{x}}))$ and $\boldsymbol{\tau}^j(\tilde{\boldsymbol{x}}) = \boldsymbol{\tau}^j(\boldsymbol{M}(\tilde{\boldsymbol{x}}))$ by (37). Then we can apply the Taylor expansion to the velocity appearing in the term $\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{n}(\tilde{\boldsymbol{x}})$, so that, again for the point $\tilde{\boldsymbol{x}}$ on the surrogate boundary $\tilde{\Gamma}_N$,

$$\begin{aligned} \boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) &= \left(\left(\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{n}\right)\boldsymbol{n} + \left(\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{\tau}^j\right)\boldsymbol{\tau}^j\right) \cdot \tilde{\boldsymbol{n}} \\ &\approx \left(\left((\boldsymbol{v}(\boldsymbol{x}) - \nabla \boldsymbol{v}(\tilde{\boldsymbol{x}})\boldsymbol{d}) \cdot \boldsymbol{n}\right)\boldsymbol{n} + (\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{\tau}^j)\boldsymbol{\tau}^j\right) \cdot \tilde{\boldsymbol{n}} \\ &= \left(v_N - \boldsymbol{n}^T \nabla \boldsymbol{v}\, \boldsymbol{d}\right)\boldsymbol{n} \cdot \tilde{\boldsymbol{n}} + (\boldsymbol{v} \cdot \boldsymbol{\tau}^j)\boldsymbol{\tau}^j \cdot \tilde{\boldsymbol{n}}, \end{aligned} \tag{63}$$

where $v_N = v_N(\boldsymbol{M}(\tilde{\boldsymbol{x}}))$. Equation (63) represents the surrogate Neumann boundary condition on $\tilde{\Gamma}_N$, and requires for consistency the tangential term $(\boldsymbol{v} \cdot \boldsymbol{\tau}^j)\boldsymbol{\tau}^j \cdot \tilde{\boldsymbol{n}}$, a byproduct of the decomposition of $\tilde{\boldsymbol{n}}$ in terms of $\boldsymbol{n}$ and $\boldsymbol{\tau}^j$.

5.1.3. Variational formulation

Using the approximate (shifted) Dirichlet and Neumann conditions we can modify formulation (41a)–(41b) as follows:

$$0 = (\phi\,,\ \chi\,\partial_t p)_{\tilde{\Omega}} - (\nabla\phi\,,\ \boldsymbol{v})_{\tilde{\Omega}} + \langle \phi\,,\ \widehat{\boldsymbol{v}\cdot\tilde{\boldsymbol{n}}}\rangle_{\tilde{\Gamma}}\,, \tag{64a}$$

$$0 = (\boldsymbol{\psi}\,,\ \partial_t \boldsymbol{v} - \boldsymbol{b})_{\tilde{\Omega}} - (\nabla\cdot\boldsymbol{\psi}\,,\ p)_{\tilde{\Omega}} + \langle \boldsymbol{\psi}\cdot\tilde{\boldsymbol{n}}\,,\ \hat{p}\rangle_{\tilde{\Gamma}}\,, \tag{64b}$$

where now

$$\widehat{\boldsymbol{v}\cdot\tilde{\boldsymbol{n}}} = \begin{cases} \boldsymbol{v}\cdot\tilde{\boldsymbol{n}}\,, & \text{on } \tilde{\Gamma}_D\,, \\ \left(v_N - \boldsymbol{n}^T(\nabla\boldsymbol{v})\boldsymbol{d}\right)\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + (\boldsymbol{v}\cdot\boldsymbol{\tau}^j)\boldsymbol{\tau}^j\cdot\tilde{\boldsymbol{n}}\,, & \text{on } \tilde{\Gamma}_N\,, \end{cases} \tag{65a}$$

$$\hat{p} = \begin{cases} p_D - \nabla p\cdot\boldsymbol{d}\,, & \text{on } \tilde{\Gamma}_D\,, \\ p\,, & \text{on } \tilde{\Gamma}_N\,. \end{cases} \tag{65b}$$

Hence, upon substitution, we obtain the variational formulation

$$0 = (\phi\,,\ \chi\,\partial_t p)_{\tilde{\Omega}} - (\nabla\phi\,,\ \boldsymbol{v})_{\tilde{\Omega}} + \langle \phi\,,\ ((v_N - \boldsymbol{n}^T(\nabla\boldsymbol{v})\boldsymbol{d})\boldsymbol{n} + (\boldsymbol{v}\cdot\boldsymbol{\tau}^j)\boldsymbol{\tau}^j)\cdot\tilde{\boldsymbol{n}}\rangle_{\tilde{\Gamma}_N} + \langle \phi\,,\ \boldsymbol{v}\cdot\tilde{\boldsymbol{n}}\rangle_{\tilde{\Gamma}_D}\,, \tag{66a}$$

$$0 = (\boldsymbol{\psi}\,,\ \partial_t\boldsymbol{v} - \boldsymbol{b})_{\tilde{\Omega}} - (\nabla\cdot\boldsymbol{\psi}\,,\ p)_{\tilde{\Omega}} + \langle \boldsymbol{\psi}\cdot\tilde{\boldsymbol{n}}\,,\ p_D - \nabla p\cdot\boldsymbol{d}\rangle_{\tilde{\Gamma}_D} + \langle \boldsymbol{\psi}\cdot\tilde{\boldsymbol{n}}\,,\ p\rangle_{\tilde{\Gamma}_N}\,. \tag{66b}$$

The previous equations can be compared with (43a)–(43b) to highlight differences with respect to the case of conformal grids. Integrating by parts in space leads to the Euler–Lagrange equations,

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

$$0 = (\phi\,,\, \chi\,\partial_t p + \nabla\cdot \boldsymbol{v})_{\tilde\Omega} - \langle (\tilde{\boldsymbol{n}}\cdot\boldsymbol{n})\,\phi\,,\, \boldsymbol{v} + (\nabla \boldsymbol{v})\,\boldsymbol{d})\cdot\boldsymbol{n} - v_N\rangle_{\tilde\Gamma_N}\,, \tag{67a}$$

$$0 = (\boldsymbol{\psi}\,,\, \partial_t \boldsymbol{v} + \nabla p - \boldsymbol{b})_{\tilde\Omega} - \langle \boldsymbol{\psi}\cdot\tilde{\boldsymbol{n}}\,,\, p - p_D + \nabla p\cdot\boldsymbol{d}\rangle_{\tilde\Gamma_D} \tag{67b}$$

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:

$$0 = (\boldsymbol{W}\,,\, \boldsymbol{A}_0\partial_t\boldsymbol{Y} - \boldsymbol{Z})_{\tilde\Omega} - (\partial_{x_i}\boldsymbol{W}\,,\, \boldsymbol{A}_i\boldsymbol{Y})_{\tilde\Omega} + \langle \boldsymbol{W}\,,\, \widehat{\boldsymbol{F}_i\tilde n_i}\rangle_{\tilde\Gamma}\,, \tag{68}$$

where

$$\widehat{\boldsymbol{F}_i\tilde n_i} = \begin{cases} \Big(\big(v_N - \boldsymbol{n}^T(\nabla\boldsymbol{v})\,\boldsymbol{d}\big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + (\boldsymbol{v}\cdot\boldsymbol{\tau}^j)\,\boldsymbol{\tau}^j\cdot\tilde{\boldsymbol{n}}\Big)\begin{Bmatrix}1\\ \boldsymbol{0}_{n_d\times 1}\end{Bmatrix} + \begin{Bmatrix}0\\ p\tilde{\boldsymbol{n}}\end{Bmatrix}, & \text{on } \tilde\Gamma_N\,,\\ \boldsymbol{v}\cdot\tilde{\boldsymbol{n}}\begin{Bmatrix}1\\ \boldsymbol{0}_{n_d\times 1}\end{Bmatrix} + (p_D - \nabla p\cdot\boldsymbol{d})\begin{Bmatrix}0\\ \tilde{\boldsymbol{n}}\end{Bmatrix}\,, & \text{on } \tilde\Gamma_D\,. \end{cases} \tag{69}$$

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

$$\tilde{\boldsymbol{H}} = \widehat{\boldsymbol{F}_i\tilde n_i} = \begin{cases} \boldsymbol{n}\cdot\tilde{\boldsymbol{n}}\left(v_N\begin{Bmatrix}1\\ \boldsymbol{0}_{n_d\times 1}\end{Bmatrix} - n_i\boldsymbol{A}_i^{\boldsymbol{v}}\nabla\boldsymbol{Y}\boldsymbol{d}\right) + \boldsymbol{\tau}^j\cdot\tilde{\boldsymbol{n}}\left(\boldsymbol{F}_i^{\boldsymbol{v}}\tau_i^j\right) + \boldsymbol{F}_i^p\tilde n_i\,, & \text{on } \tilde\Gamma_N\,,\\ p_D\begin{Bmatrix}0\\ \tilde{\boldsymbol{n}}\end{Bmatrix} - \tilde n_i\boldsymbol{A}_i^p\nabla\boldsymbol{Y}\boldsymbol{d} + \boldsymbol{F}_i^{\boldsymbol{v}}\tilde n_i\,, & \text{on } \tilde\Gamma_D\,, \end{cases} \tag{70}$$

in which $\boldsymbol{A}_i^{\boldsymbol{v}} = \boldsymbol{F}_{i\,,\boldsymbol{Y}}^{\boldsymbol{v}}$ and $\boldsymbol{A}_i^p = \boldsymbol{F}_{i\,,\boldsymbol{Y}}^p$, so that $\nabla\boldsymbol{F}_i^{\boldsymbol{v}}\boldsymbol{d} = \boldsymbol{A}_i^{\boldsymbol{v}}\nabla\boldsymbol{Y}\boldsymbol{d}$ and $\nabla\boldsymbol{F}_i^p\boldsymbol{d} = \boldsymbol{A}_i^p\nabla\boldsymbol{Y}\boldsymbol{d}$, respectively. Then the vector form of (66a)–(66b) is expressed as

$$0 = (\boldsymbol{W}\,,\, \boldsymbol{A}_0\partial_t\boldsymbol{Y})_{\tilde\Omega} - (\partial_{x_i}\boldsymbol{W}\,,\, \boldsymbol{A}_i\boldsymbol{Y})_{\tilde\Omega} - (\boldsymbol{W}\,,\, \boldsymbol{Z})_{\tilde\Omega} + \langle\boldsymbol{W}\,,\, \tilde{\boldsymbol{H}}\rangle_{\tilde\Gamma} + (\mathscr{L}\boldsymbol{W}\,,\, \tau\,\boldsymbol{A}_0^{-1}\,(\mathscr{L}_t\boldsymbol{Y} - \boldsymbol{Z}))_{\tilde\Omega}\,. \tag{71}$$

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 $\tilde{\boldsymbol{H}}$, 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 $\tilde{\boldsymbol{H}}$ 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

$$\frac{1}{2}g\,h^2(\tilde{\boldsymbol{x}}) \approx \frac{1}{2}g\,\Big(\big(\eta_*(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\big)\Big)^2\,, \tag{72}$$

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

$$\tilde{\boldsymbol{H}}_*^h(\tilde{\boldsymbol{x}}) = \frac{1}{2}g\,\big(\eta_*(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\big)^2\begin{Bmatrix}0\\ \tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}})\end{Bmatrix}\,. \tag{73}$$

The idea behind the specific form of the boundary conditions on $\tilde\Gamma_D$ and $\tilde\Gamma_{I;sup}$ is that since h is extrapolated using the Taylor expansion from the surrogate to the true boundary, the same need to hold for the bathymetry z, in order to maintain a constant free surface condition. This means that the given bathymetry z is approximated by the extrapolated bathymetry in the region between true and surrogate boundaries. This will have important consequences of the preservation of a constant free surface boundary condition, as in the case of a lake at rest. More details will be given momentarily.

Turning now our attention to the term $\tilde{\boldsymbol{H}}_N^{\boldsymbol{v}}$ on Γ_N, its design is rather straightforward using (63) and other results already obtained in the acoustics case. A term that requires instead special care is the term $\tilde{\boldsymbol{H}}_*^{\boldsymbol{v}}$, on the surrogate of $\Gamma_* = \Gamma_{O;sub}$, where we impose $h\,\boldsymbol{v}\cdot\boldsymbol{n} = m_*$. We start from (63) and derive, neglecting quadratic and higher-order terms:

$$
\begin{aligned}
h(\tilde{\boldsymbol{x}})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) &= h(\tilde{\boldsymbol{x}})\,\Big(\big(\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n}\big)\,\boldsymbol{n} + \big(\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{\tau}\big)\,\boldsymbol{\tau}\Big)\cdot\tilde{\boldsymbol{n}} \\
&\approx h(\tilde{\boldsymbol{x}})\,\Big(\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) - \boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(h(\tilde{\boldsymbol{x}})\,\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(h(\boldsymbol{x})\,\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&= \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,(\boldsymbol{v}(\tilde{\boldsymbol{x}}) + (\nabla\boldsymbol{v})\boldsymbol{d})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}\,,
\end{aligned} \tag{74}
$$

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

$$
\tilde{\boldsymbol{H}}^{\boldsymbol{v}}_*(\tilde{\boldsymbol{x}}) = \left(\Big(m_* - (\nabla h\cdot\boldsymbol{d})\,\boldsymbol{v}\cdot\boldsymbol{n} - h\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}\right)\begin{Bmatrix} 1 \\ \boldsymbol{v} \end{Bmatrix}\,, \tag{75}
$$

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

In the case of a subcritical inlet $\Gamma_* = \Gamma_{I;sub}$, one additional boundary condition $\boldsymbol{v}\cdot\boldsymbol{\tau} = 0$ on the tangential component of velocity needs to be imposed on the surrogate boundary. We start from (74) but we consider a revised approximation for $\boldsymbol{v}\cdot\boldsymbol{\tau} = 0$, in the term

$$
\begin{aligned}
h(\tilde{\boldsymbol{x}})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) &\approx \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{\tau}(\boldsymbol{x}) - (\boldsymbol{\tau})^T\nabla\boldsymbol{v}\,\boldsymbol{d})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&\approx \Big(m_* - (\nabla h(\tilde{\boldsymbol{x}})\cdot\boldsymbol{d})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\boldsymbol{n}(\boldsymbol{x}) - h(\tilde{\boldsymbol{x}})\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h(\tilde{\boldsymbol{x}})\,(-(\boldsymbol{\tau})^T\nabla\boldsymbol{v}\,\boldsymbol{d})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}} \\
&= \tilde{m}_*\,,
\end{aligned} \tag{76}
$$

$$
\begin{aligned}
h(\tilde{\boldsymbol{x}})\,\boldsymbol{v}(\tilde{\boldsymbol{x}})\cdot\tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}})\,\boldsymbol{v}(\tilde{\boldsymbol{x}}) &\approx \tilde{m}_*\,(\boldsymbol{v}(\boldsymbol{x}) - \nabla\boldsymbol{v}\,\boldsymbol{d}) \\
&\approx \tilde{m}_*\,((\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n})\,\boldsymbol{n} + (\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{\tau})\,\boldsymbol{\tau} - \nabla\boldsymbol{v}\,\boldsymbol{d}) \\
&\approx \tilde{m}_*\,((\boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n})\,\boldsymbol{n} - \nabla\boldsymbol{v}\,\boldsymbol{d}) \\
&\approx \tilde{m}_*\,\big(((\boldsymbol{v}(\tilde{\boldsymbol{x}}) + \nabla\boldsymbol{v}\,\boldsymbol{d})\cdot\boldsymbol{n})\,\boldsymbol{n} - \nabla\boldsymbol{v}\,\boldsymbol{d}\big)\,.
\end{aligned} \tag{77}
$$

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

$$
\tilde{\boldsymbol{H}}^{\boldsymbol{v}}_*(\tilde{\boldsymbol{x}}) = \left(\Big(m_* - (\nabla h\cdot\boldsymbol{d})\,\boldsymbol{v}\cdot\boldsymbol{n} - h\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\Big)\,\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h\,(-(\boldsymbol{\tau})^T\,\nabla\boldsymbol{v}\,\boldsymbol{d})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}\right)\begin{Bmatrix} 1 \\ ((\boldsymbol{v}(\tilde{\boldsymbol{x}}) + \nabla\boldsymbol{v}\,\boldsymbol{d})\cdot\boldsymbol{n})\,\boldsymbol{n} - \nabla\boldsymbol{v}\,\boldsymbol{d} \end{Bmatrix}\,, \tag{78}
$$

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

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

$$
\tilde{\boldsymbol{H}}^{\boldsymbol{v}}_{I;sup}(\tilde{\boldsymbol{x}}) = \big(\eta_{I;sup}(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\big)\,\big(\boldsymbol{v}_{I;sup} - (\nabla\boldsymbol{v})\boldsymbol{d}\big)\begin{Bmatrix} 1 \\ \boldsymbol{v}_{I;sup} - (\nabla\boldsymbol{v})\boldsymbol{d} \end{Bmatrix}\,. \tag{79}
$$

In conclusion, we obtain the following final variational statement

$$
0 = (\boldsymbol{W}\,,\,\partial_t\boldsymbol{U} - \boldsymbol{Z})_{\tilde{\Omega}} - \big(\partial_{x_i}\boldsymbol{W}\,,\,\boldsymbol{F}_i\big)_{\tilde{\Omega}} + \langle\boldsymbol{W}\,,\,\tilde{\boldsymbol{H}}\rangle_{\tilde{\Gamma}} + (\mathscr{L}^T\boldsymbol{W}\,,\,\tau\,\hat{\boldsymbol{A}}_0^{-1}\,(\mathscr{L}_t\boldsymbol{Y} - \boldsymbol{Z}))_{\tilde{\Omega}} + (\partial_{x_i}\boldsymbol{W}\,,\,\nu\,g^{ij}\,\hat{\boldsymbol{A}}_0\,\partial_{x_j}\boldsymbol{Y})_{\tilde{\Omega}}\,, \tag{80a}
$$

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

$$\tilde{\boldsymbol{H}}^{\boldsymbol{v}}(\boldsymbol{Y}) = \begin{cases} \left(\left(v_N - \boldsymbol{n}^T \nabla \boldsymbol{v}\, \boldsymbol{d}\right) \boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + (\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}\right) \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \boldsymbol{v}\cdot\tilde{\boldsymbol{n}} \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_D, \\ \tilde{m}_{I;sub} \begin{Bmatrix} 1 \\ ((\boldsymbol{v}(\tilde{\boldsymbol{x}}) + \nabla \boldsymbol{v}\,\boldsymbol{d})\cdot\boldsymbol{n})\,\boldsymbol{n} - \nabla\boldsymbol{v}\,\boldsymbol{d} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \left(\eta_{I;sup}(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\right)\left(\boldsymbol{v}_{I;sup} - (\nabla\boldsymbol{v})\boldsymbol{d}\right) \begin{Bmatrix} 1 \\ \boldsymbol{v}_{I;sup} - (\nabla\boldsymbol{v})\boldsymbol{d} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ \tilde{m}_{O;sub} \begin{Bmatrix} 1 \\ \boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \boldsymbol{v}\cdot\tilde{\boldsymbol{n}} \begin{Bmatrix} h \\ h\,\boldsymbol{v} \end{Bmatrix}, & \text{on } \Gamma_{O;sup} \end{cases} \tag{80b}$$

and

$$\tilde{\boldsymbol{H}}^{h}(\boldsymbol{Y}) = \begin{cases} \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_N, \\ \frac{1}{2} g \left(\eta_D(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\right)^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_D, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_{I;sub}, \\ \frac{1}{2} g \left(\eta_{I;sup}(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\right)^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_{I;sup}, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_{O;sub}, \\ \frac{1}{2} g h^2 \begin{Bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{Bmatrix}, & \text{on } \Gamma_{O;sup}, \end{cases} \tag{80c}$$

where

$$\tilde{m}_{I;sub} = \left(m_{I;sub} - (\nabla h\cdot\boldsymbol{d})\,\boldsymbol{v}\cdot\boldsymbol{n} - h\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\right)\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h\,(-(\boldsymbol{\tau})^T\,\nabla\boldsymbol{v}\,\boldsymbol{d})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}, \tag{80d}$$

$$\tilde{m}_{O;sub} = \left(m_{O;sub} - (\nabla h\cdot\boldsymbol{d})\,\boldsymbol{v}\cdot\boldsymbol{n} - h\,\boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\right)\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} + h\,(\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}. \tag{80e}$$

Similar to the case of body-fitted conformal grids, also in the case of the SB method penalty terms are added to enhance the stability and convergence properties of the open sea and wall boundary conditions:

$$\langle \boldsymbol{\psi}\cdot\tilde{\boldsymbol{n}} + \left(\boldsymbol{n}^T\nabla\boldsymbol{\psi}\,\boldsymbol{d}\right)\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} - (\boldsymbol{\psi}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}},\, \alpha h\left(\boldsymbol{v}\cdot\tilde{\boldsymbol{n}} - \left(v_N - \boldsymbol{n}^T\nabla\boldsymbol{v}\,\boldsymbol{d}\right)\boldsymbol{n}\cdot\tilde{\boldsymbol{n}} - (\boldsymbol{v}\cdot\boldsymbol{\tau})\,\boldsymbol{\tau}\cdot\tilde{\boldsymbol{n}}\right)\rangle_{\tilde{\Gamma}_N}, \quad \text{on } \Gamma_N, \tag{81a}$$

$$\langle \phi + \nabla\phi\cdot\boldsymbol{d},\, \alpha\,(h - (\eta_D - z) + \nabla h\cdot\boldsymbol{d})\rangle_{\tilde{\Gamma}_D}, \quad \text{on } \Gamma_D. \tag{81b}$$

5.3. Discrete conservation properties

Also in the case of the SB method, a statement of conservation can be derived for the variational formulations (71) and (80), by using the same test function $\boldsymbol{W} = \boldsymbol{1}_j$ defined in Section 4.3, but this time supported over $\tilde{\Omega}$ only. Namely, for (80), we have,

$$0 = (\boldsymbol{1}_j\,,\, \partial_t\boldsymbol{U} - \boldsymbol{Z})_{\tilde{\Omega}} + \langle \boldsymbol{1}_j\,,\, \tilde{\boldsymbol{H}}\rangle_{\tilde{\Gamma}}, \tag{82}$$

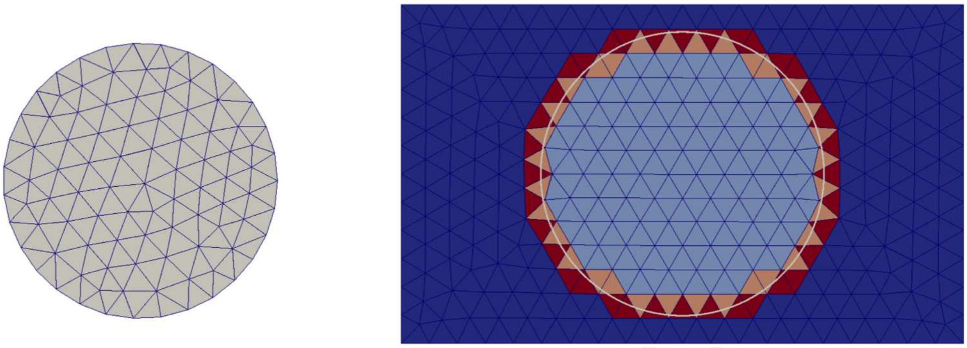


Fig. 3. Linear acoustic wave convergence test. Computational domain geometry and grids at the coarsest level of refinement: conformal grid (left) and embedded grid (right). For the embedded grid, light blue indicates $\tilde{\Omega}$, that is the active elements inside the computational domain, dark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

and a similar statement holds for (71). Observe that the conservation statement is with respect to the numerical fluxes $\tilde{\boldsymbol{H}}$ and not $\boldsymbol{H}$. $\tilde{\boldsymbol{H}}$ and $\boldsymbol{H}$ are related by the Taylor expansion used to derive the surrogate boundary conditions.

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 $\boldsymbol{F}_i^h n_i$ and $\tilde{\boldsymbol{H}}$ in the expression

$$(\boldsymbol{W}\,,\,\partial_t \boldsymbol{U})_{\tilde{\Omega}} = \left(\boldsymbol{W}\,,\,\boldsymbol{Z}_o - \partial_{x_i}\boldsymbol{F}_i\right)_{\tilde{\Omega}} + \langle \boldsymbol{W}\,,\,\boldsymbol{F}_i^h n_i - \tilde{\boldsymbol{H}}\rangle_{\tilde{\Gamma}}\,, \tag{83}$$

analogous to (60), may not always simplify exactly. Comparing with (80b)–(80c), 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.

Verifying that cancellations also take place on the boundaries $\tilde{\Gamma}_D$ and $\tilde{\Gamma}_{I;sup}$ takes a little more effort. Recalling that the given bathymetry z is approximated by the extrapolated bathymetry in the region between true and surrogate boundaries, we have that $[\nabla(h+z)](\tilde{\boldsymbol{x}}) = \nabla\eta_0 = 0$ in the terms associated with $\tilde{\boldsymbol{H}}^h$, and consequently:

$$\frac{1}{2} g \left(\eta_0(\boldsymbol{x}) - ([\nabla(h+z)](\tilde{\boldsymbol{x}}))\cdot\boldsymbol{d} - z(\tilde{\boldsymbol{x}})\right)^2 = \frac{1}{2} g \left(\eta_0(\boldsymbol{x}) - z(\tilde{\boldsymbol{x}})\right)^2 = \frac{1}{2} g \left(h(\tilde{\boldsymbol{x}})\right)^2\,, \tag{84}$$

which cancels exactly the terms associated with $\boldsymbol{F}_i^h n_i$.

6. Time integration

The integration in time of the equations is performed using the same algorithms presented in [47,50] for body-fitted conformal grids, to which the reader can refer for more details and the analysis of stability and convergence. 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.

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

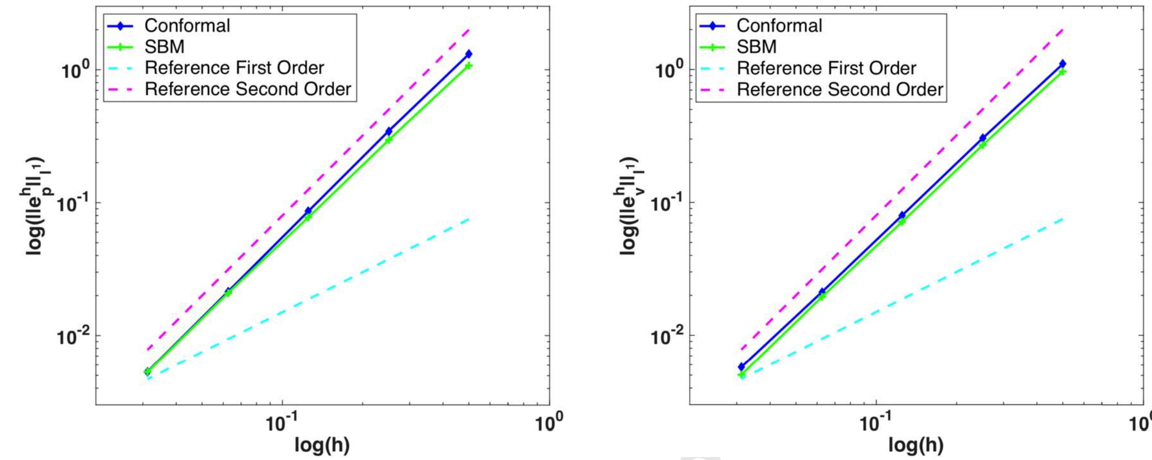


Fig. 4. Convergence of the $l^1([0,T]:(L^2(\Omega)))$-norm for the acoustic pulse propagation problem on an embedded disk domain with zero normal velocity boundary conditions. On the left, pressure, and on the right, velocity. A comparison between SB method and the conformal Nitsche method is shown for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

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 Fig. 3, with corresponding conformal and embedded grids. The exact solution is radially symmetric and given by

$$v_r(\boldsymbol{r}, t) = \frac{1}{\rho\, c_s} J_1\!\left(\frac{z_{1,1}}{R}\boldsymbol{r}\right) \sin\left(\frac{c_s\, z_{1,1}}{R} t\right),$$

$$v_\theta(\boldsymbol{r}, t) = 0,$$

$$p(\boldsymbol{r}, t) = J_0\!\left(\frac{z_{1,1}}{R}\boldsymbol{r}\right) \cos\left(\frac{c_s\, z_{1,1}}{R} t\right),$$

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.

Fig. 4 shows the convergence of the pressure and velocity for this test, in the $l^1([0,T]:(L^2(\Omega)))$-norm, defined as:

$$\|\boldsymbol{e}\|_{l^1([0,T]:(L^2(\Omega^*)))} = \frac{1}{N+1} \sum_{i=0}^{N+1} \|\boldsymbol{e}(t_i)\|_{L^2(\Omega^*)}, \tag{85}$$

where $\Omega^* = \Omega$ or $\tilde{\Omega}$, $t_0 = 0$ is the initial time of the simulation, and $t_{N+1} = T$ is the final time of the simulation. In what follows, we will abbreviate the notation $l^1([0,T]:(L^2(\Omega^*)))$-norm to $l^1([0,T]:L^2)$-norm. 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 Fig. 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,

$$v_r(\boldsymbol{r}, t) = \frac{1}{\rho\, c_s} J_1\!\left(\frac{z_{0,1}}{R}\boldsymbol{r}\right) \sin\left(\frac{c_s\, z_{0,1}}{R} t\right),$$

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

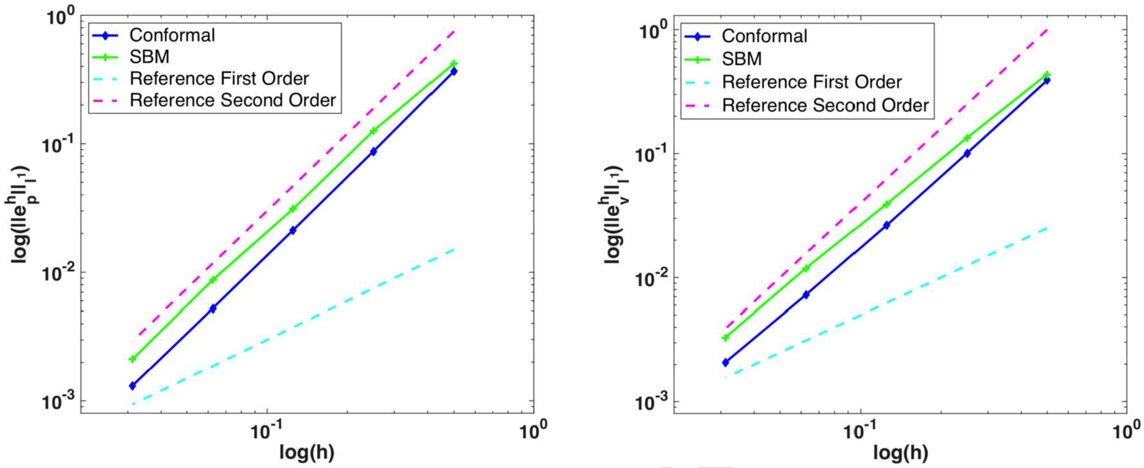


Fig. 5. Convergence of the $l^1([0,T]:(L^2(\Omega)))$-norm for the acoustic pulse propagation problem on an embedded disk domain with zero pressure boundary conditions. On the left, pressure, and on the right, velocity. A comparison between SB method and the conformal Nitsche method is shown for the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

$$v_\theta(\boldsymbol{r}, t) = 0\,,$$

$$p(\boldsymbol{r}, t) = J_0\left(\frac{z_{0,1}}{R}\boldsymbol{r}\right)\cos\left(\frac{c_s\, z_{0,1}}{R}t\right),$$

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 compute the error of pressure and velocity in the $l^1([0,T]:(L^2(\Omega)))$ norm in Fig. 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.

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:

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

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

where $\mathrm{A} = 0.01$ 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.

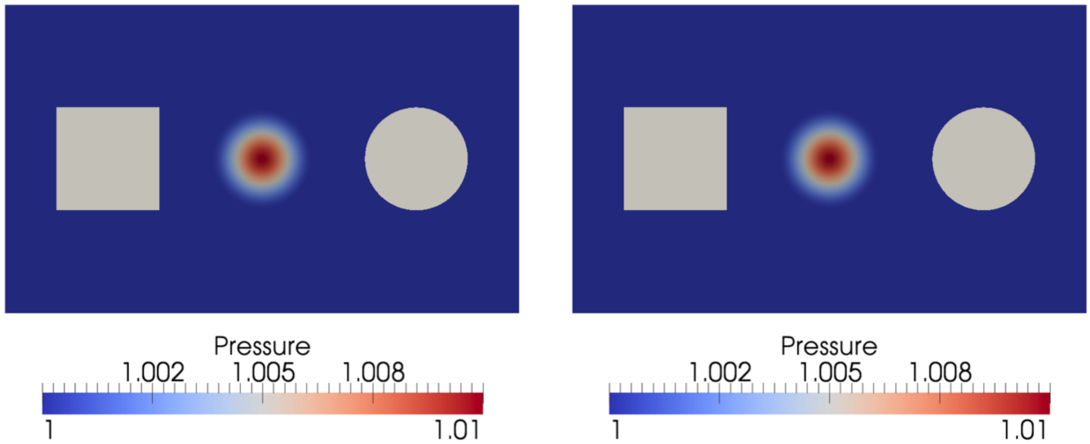
The computation using the SB method was performed on a grid that filled the entire rectangular domain $[-5,5]\times[-3,3]$ and of element size approximately 0.0312. In this test case, it is easy to observe that the solution is free from spurious oscillations throughout the many reverberations of the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

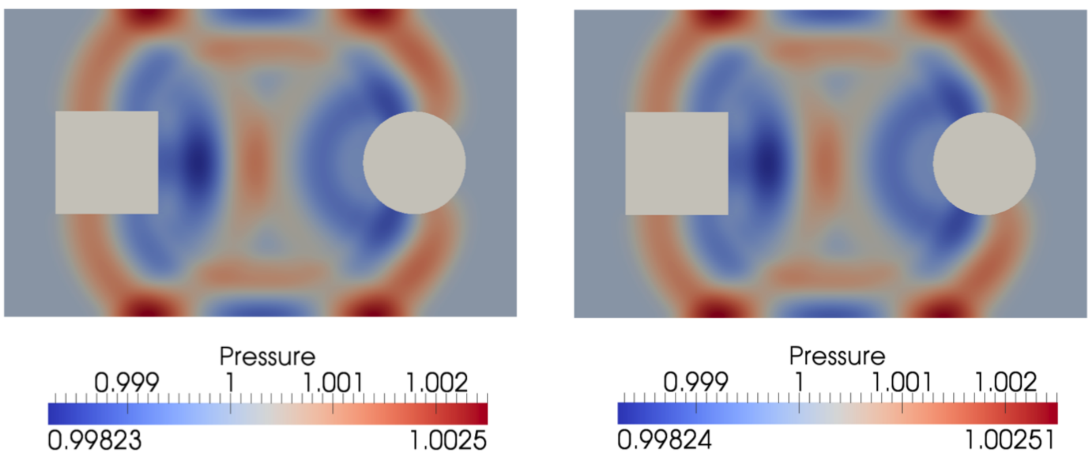
An overlay of the results near the rectangular hole from the SB and conformal Nitsche methods is shown in Fig. 8, where the solution of the SB method is shown as a solid surface and the solution of the conformal Nitsche method as a black wireframe. The two results show fairly good agreement.

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

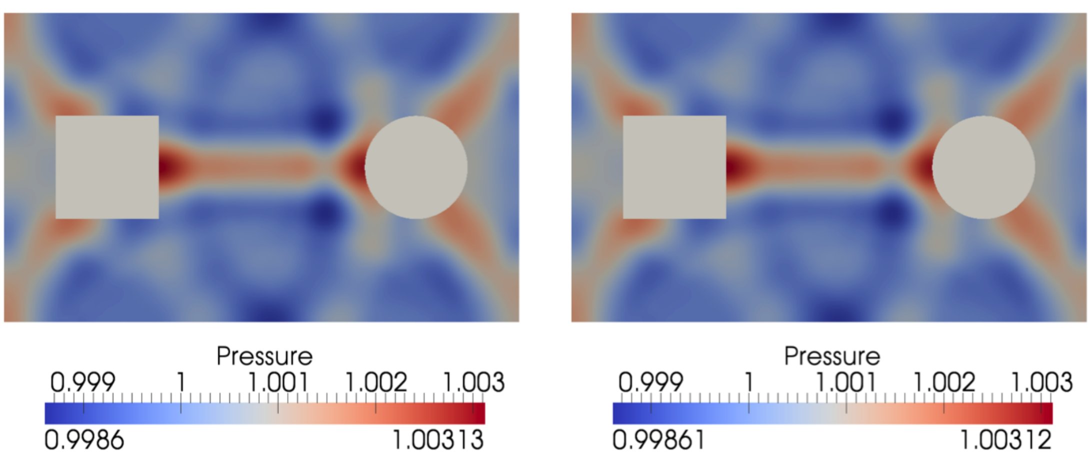
JID:YJCPH AID:7992 /FLA [m3G; v1.235; Prn:5/05/2018; 17:18] P.21 (1-35)



(a) $t = 0.0$: Conformal Nitsche method (left) and SB method (right).

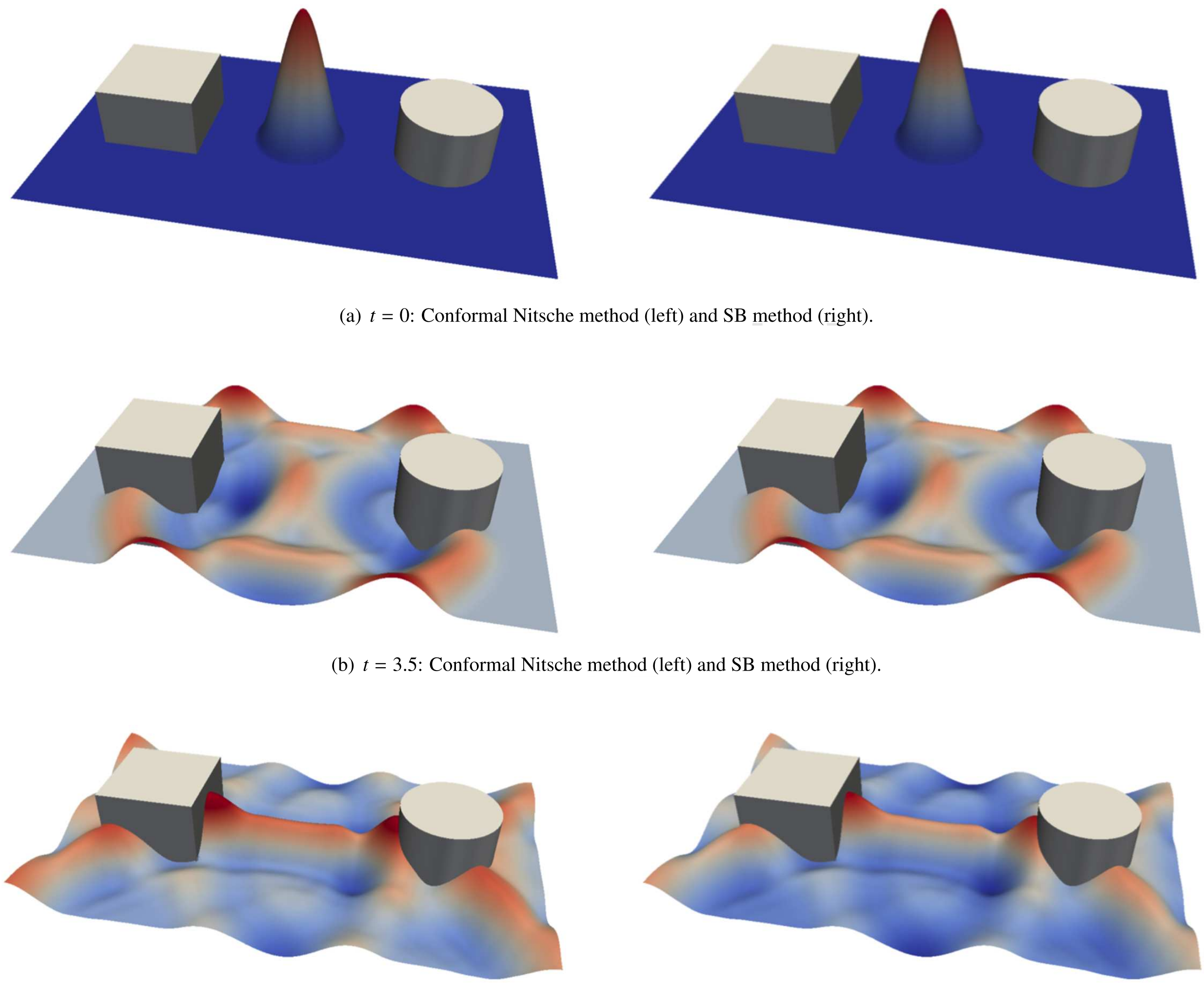


(b) $t = 3.5$: Conformal Nitsche method (left) and SB method (right).



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

Fig. 6. Comparison between pressure contours obtained with the conformal and SB methods for the propagation of an acoustics 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).

Fig. 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.

8. Numerical results for shallow water flows

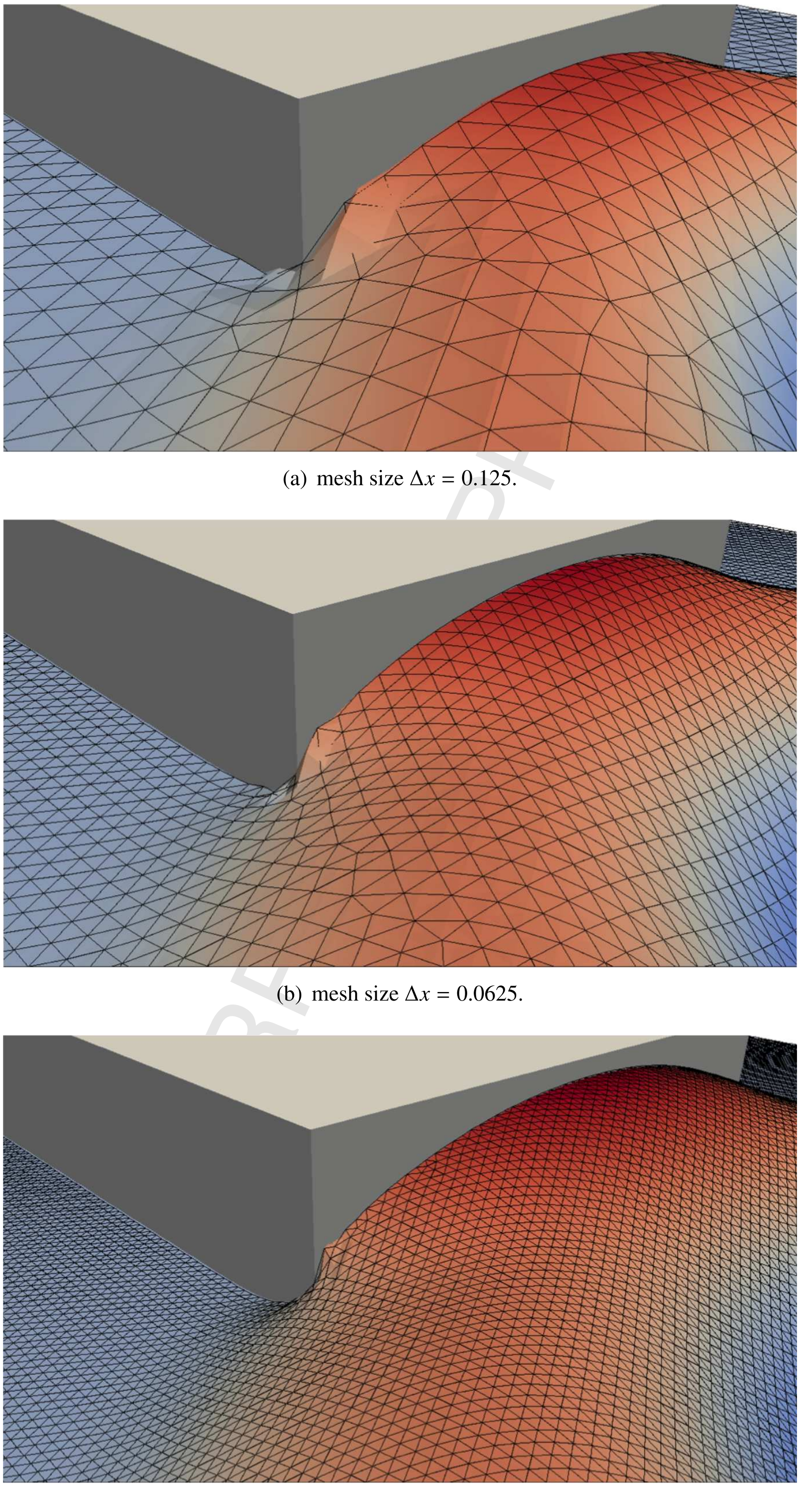
Also in the case of shallow water flows, we compare the performance of the proposed SB method against a reference Nitsche method on conformal grids of equivalent resolution. All computations were run with four corrector passes of the second-order Petrov–Galerkin space–time integrator proposed in [50] (very similar to RK2 explicit integrator) with a Courant–Lewy–Friedrichs (CFL) condition of 0.5. Moreover, we chose the stabilization parameter $c_\tau = 0.5$, and the primary variables $\boldsymbol{Y} = \{h, \boldsymbol{v}^T\}^T$ as solution variables. The penalty terms detailed in (57) and (81) for the wall and open sea boundary conditions are employed with $\alpha = 2.0$ in some cases, as specifically indicated in the following. 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

$$z(x, y) = 0.25\,(x - x_{min})\ . \tag{86}$$

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



(a) mesh size $\Delta x = 0.125$.

(b) mesh size $\Delta x = 0.0625$.

(c) mesh size $\Delta x = 0.0312$.

Fig. 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.

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.

	SB method		*Conformal Nitsche method*	
	height	velocity	height	velocity
Constant slope bottom	1.33205e−13	3.42418e−13	4.52703e−15	3.93743e−13
Central hump bottom	2.10115e−13	4.33058e−13	5.26521e−15	5.04488e−13
Cosine shape bottom	3.0125e−15	7.37541e−13	2.89032e−15	4.86065e−13

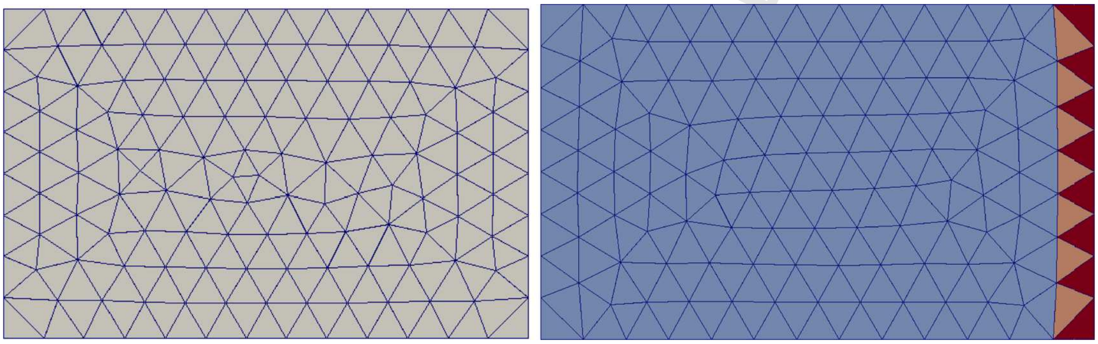


Fig. 9. Nonlinear shallow water convergence test. Computational domain geometry and grids at the coarsest level of refinement: conformal grid (left) and embedded grid (right). For the embedded grid, light blue indicates $\tilde{\Omega}$, that is the active elements inside the computational domain, and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

$$z(x, y) = \mathscr{I}_1^h \left[0.4 \exp\left(-10\left(x^2 + y^2\right)\right)\right], \tag{87}$$

where $\mathscr{I}_1^h$ 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,

$$z(x, y) = 0.25 + 0.25\, \mathscr{I}_1^h \left[cos(\pi\,(x - x_{min}))\right], \tag{88}$$

and fixed height boundary conditions are applied to the left and right hand side boundaries. Specifically, the correction suggested in (84) is used to maintain the C-property for the SB method. The other two sides are considered as solid walls. Penalty terms are included for this test, with penalty parameter $\alpha = 2.0$.

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

$$h(x, y) + z(x, y) = 1.0\,, \tag{89}$$

$$\boldsymbol{v} = \boldsymbol{0}\,, \tag{90}$$

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.

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 Fig. 9.

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

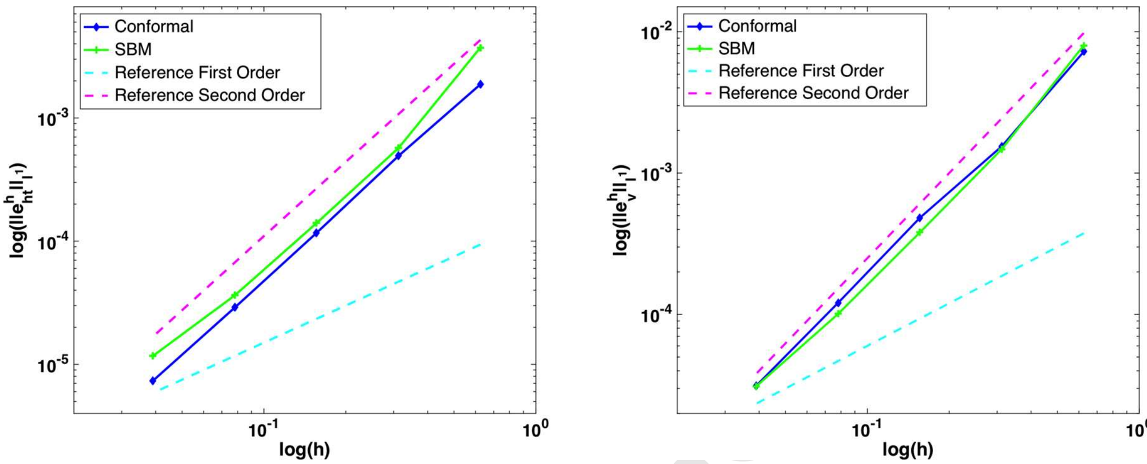


Fig. 10. Convergence of the $l^1([0,T]:(L^2(\Omega)))$-norm for the shallow water flows with zero normal velocity boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. Comparison between the SB and conformal Nitsche methods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

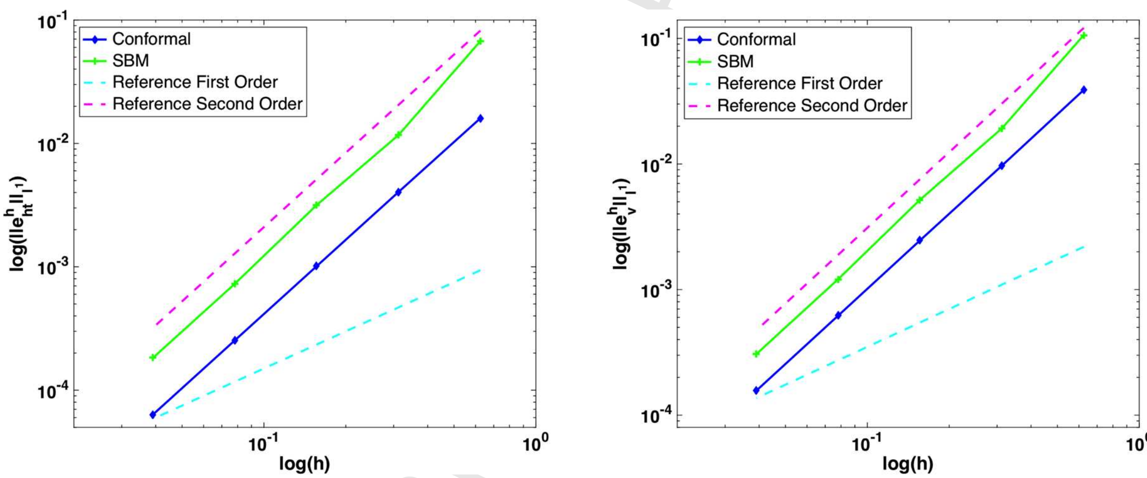


Fig. 11. Convergence of the $l^1([0,T]:(L^2(\Omega)))$-norm for the shallow water flows with an inflow boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. A comparison between the SB and the conformal Nitsche methods is shown.

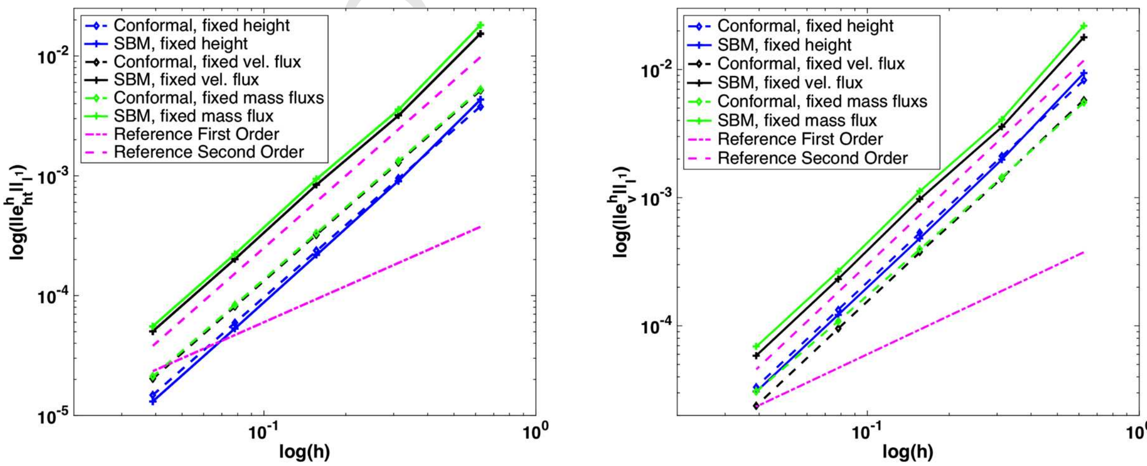
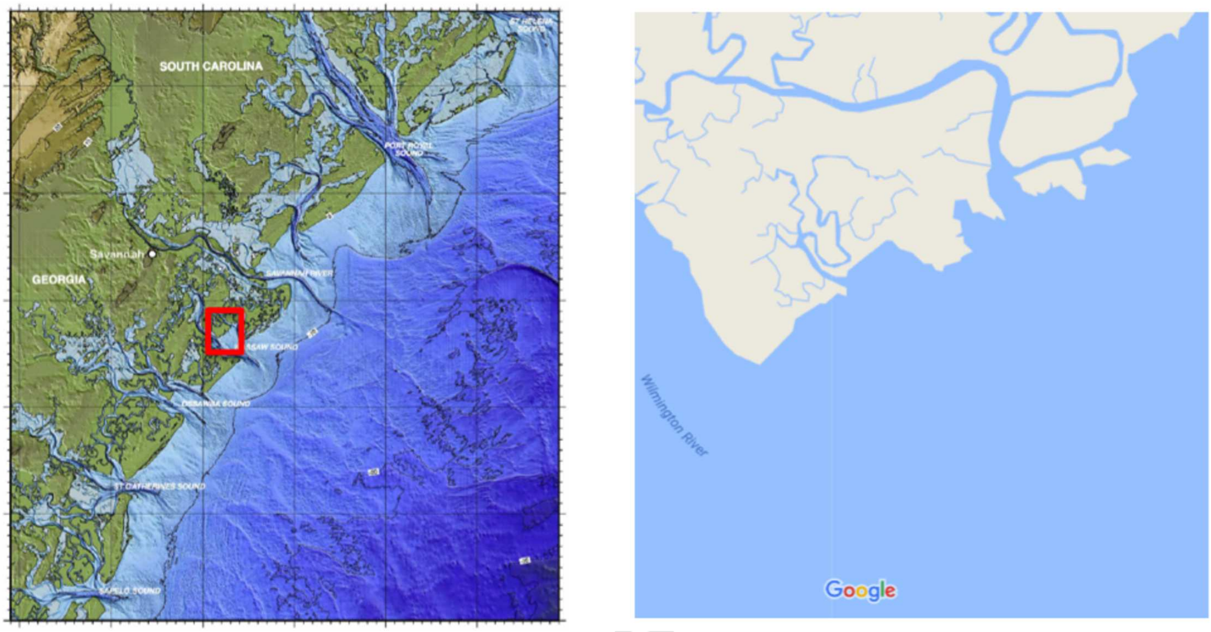
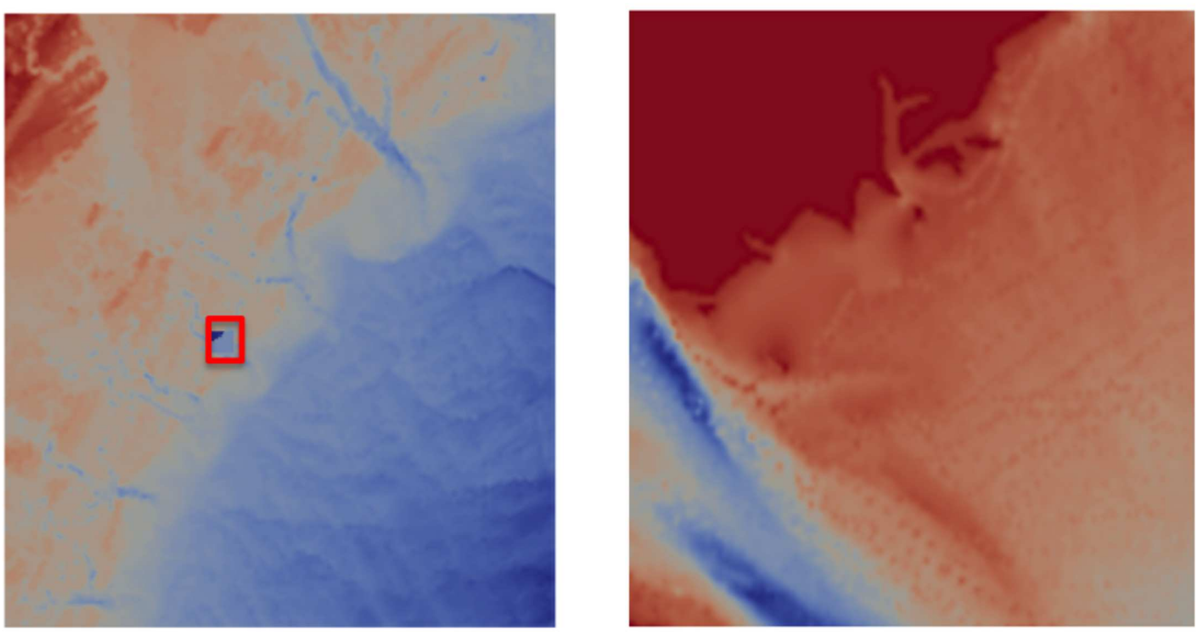


Fig. 12. Convergence of the $l^1([0,T]:(L^2(\Omega)))$-norm for the shallow water flows with outflow boundary condition on the embedded vertical right side of the rectangular domain. On the left, height, and on the right, velocity. Three different choices, namely fixed height, fixed velocity flux, and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



(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

Fig. 13. The coastline near Savannah (GA).

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

$$h = 2 + 0.1\sin(\omega t)\cos(\lambda x)\,, \tag{91a}$$

$$v_1 = 0.1\sin(\lambda x)\,, \tag{91b}$$

$$v_2 = 0\,, \tag{91c}$$

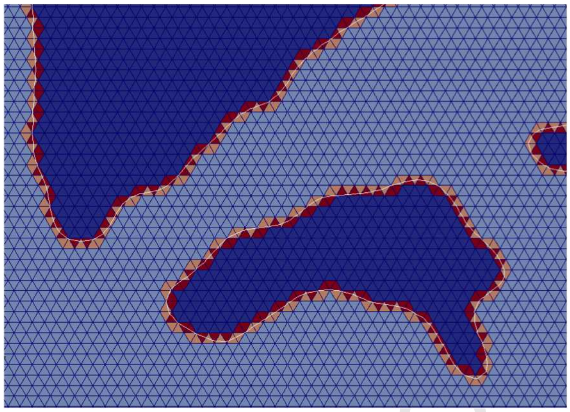


Fig. 14. A closeup of the coastline and the embedded grids utilized in the computation. White solid lines represent the zero elevation isoline. Light blue indicates the active elements inside the computational domain, dark blue indicates the elements outside, and red and orange indicate the elements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

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

$$S_1 = 0.1\,(\cos(\lambda x)\,(2\lambda + \omega\cos(\omega t)) + 0.1\lambda\cos(2\lambda x)\sin(\omega t))\,, \tag{92a}$$

$$\begin{aligned} S_2 =\ & 0.1\sin(\lambda x)\,(\lambda\,(0.005 - 2g + 0.015\cos(2\lambda x))\sin(\omega t) \\ & + 0.1\cos(\lambda x)\,(4\lambda + \omega\cos(\omega t) - g\lambda\sin(\omega t)\sin(\omega t)))\,, \end{aligned} \tag{92b}$$

$$S_3 = 0\,, \tag{92c}$$

where we set, in particular, $\lambda = \pi/L$ and $\omega = 1$. In this test, boundary penalty terms are added with penalty parameter $\alpha = 2.0$, for both the conformal and SB methods. Fig. 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 $h\,\boldsymbol{v}\cdot\boldsymbol{n} = 0.2$ is applied to the left boundary. An inflow boundary condition ($h\,\boldsymbol{v}\cdot\boldsymbol{n} = -0.2$) is applied to the right wall, and implemented as embedded in the case of the SB method. The exact solution is given by

$$h = 2 - 0.1\cos(\omega t)\sin(\lambda x)\,, \tag{93a}$$

$$v_1 = -0.1\cos(\lambda x)\,, \tag{93b}$$

$$v_2 = 0\,, \tag{93c}$$

obtained with the source terms:

$$S_1 = -0.1\,(-0.1\lambda\cos(2\lambda x)\cos(\omega t) - \sin(\lambda x)(2\lambda + \omega\sin(\omega t)))\,, \tag{94a}$$

$$\begin{aligned} S_2 =\ & -0.1\cos(\lambda x)\,(\lambda\,(-0.005 + 2g + 0.015\cos(2\lambda x))\cos(\omega t) \\ & - 0.1g\,\lambda\cos(\omega)^2\sin(\lambda x) + 0.1\sin(\lambda x)\,(4\lambda + \omega\sin(\omega t))\Big)\,, \end{aligned} \tag{94b}$$

$$S_3 = 0\,, \tag{94c}$$

where $\lambda = \pi/(2L)$ and $\omega = 1$.

Note that the inflow boundary condition is enforced using the two separate conditions

$$h\,\boldsymbol{v}\cdot\boldsymbol{n} = 0.2 \quad \text{and} \quad \boldsymbol{v}\cdot\boldsymbol{\tau} = 0\,, \tag{95}$$

as suggested by the theory of Riemann invariants. As shown in Fig. 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.

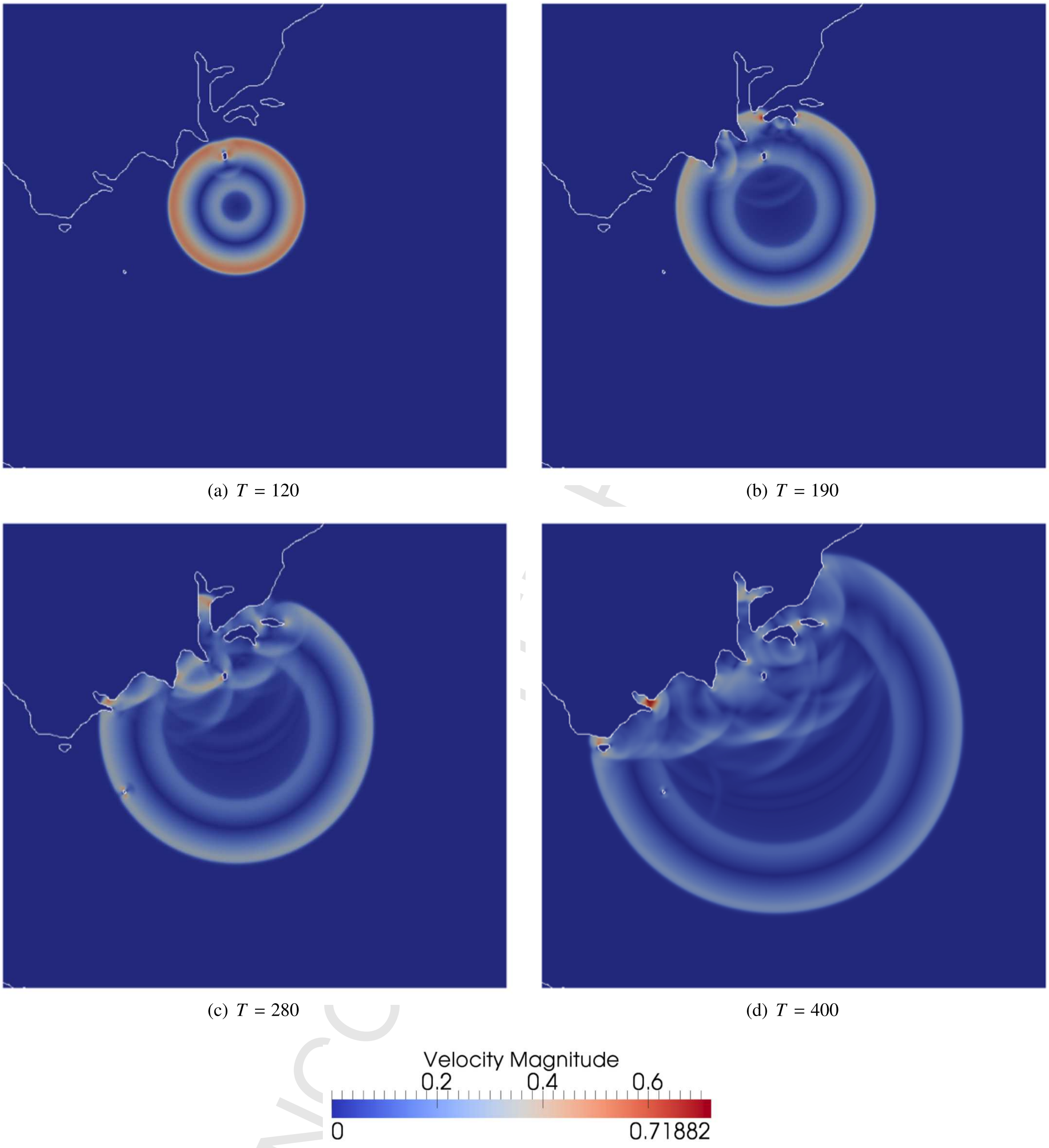


Fig. 15. Velocity magnitude contours for the propagation of a tsunami along the coastline near Savannah, (GA).

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 ($\boldsymbol{v} \cdot \boldsymbol{n} > 0$) are applied to the left and right boundaries of the rectangular domain. The exact solution is given by

$$h = 2 - 0.1 \cos(\omega t) \sin(\lambda x) ,$$
$$v_1 = -0.1 \cos(\lambda x) ,$$
$$v_2 = 0 ,$$

and is obtained with the source terms

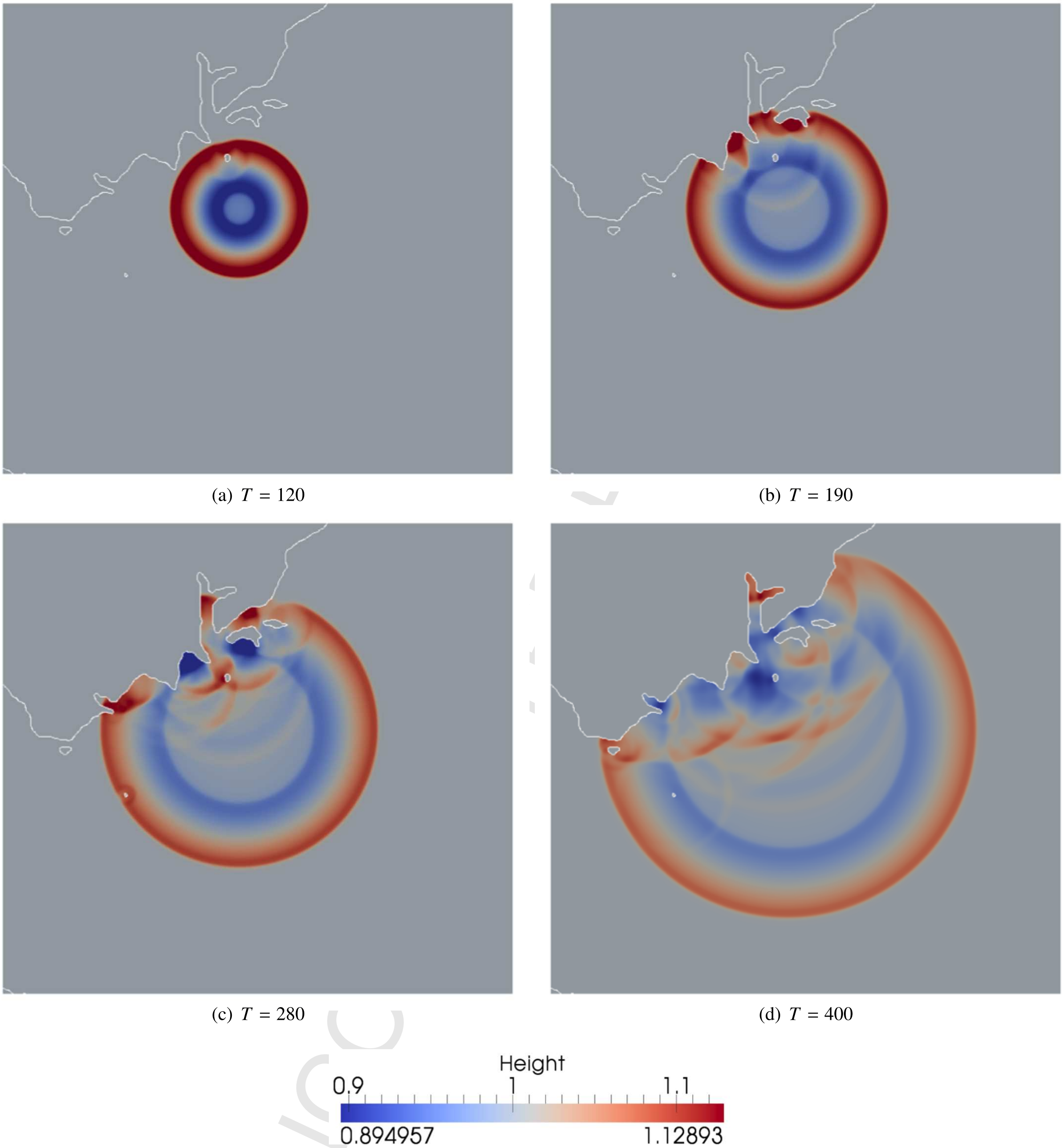


Fig. 16. Pressure contours for the propagation of a tsunami along the coastline near Savannah, (GA).

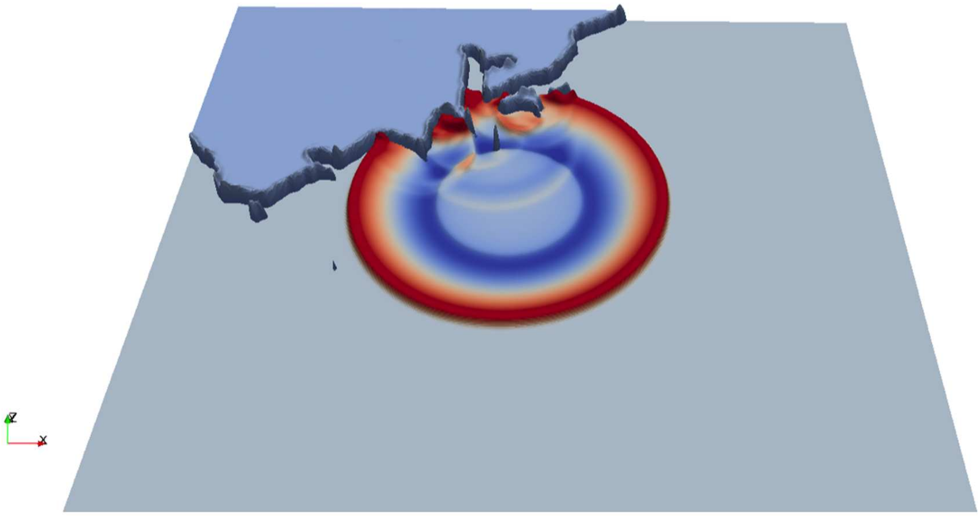
$$S_1 = -0.1\,(-0.1\lambda\,\cos(2\lambda x)\,\cos(\omega t) - \sin(\lambda x)(2\lambda + \omega\sin(\omega t)))\;, \tag{97a}$$

$$S_2 = -0.1\,\cos(\lambda x)\,(\lambda\,(-0.005 + 2g + 0.015\,\cos(2\lambda x))\,\cos(\omega t)$$
$$-0.1 g\,\lambda\cos(\omega)^2\,\sin(\lambda x) + 0.1\,\sin(\lambda x)\,(4\lambda + \omega\,\sin(\omega t))\Big)\;, \tag{97b}$$

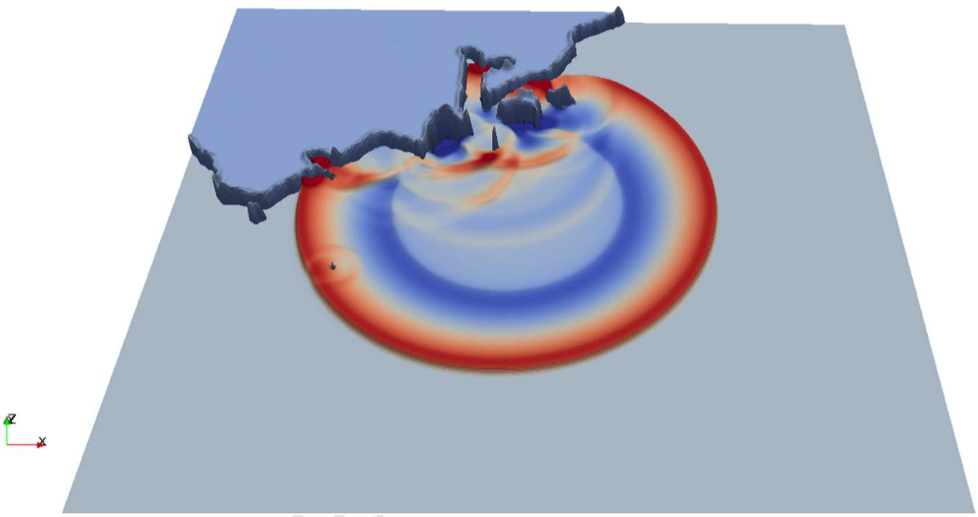
$$S_3 = 0\,, \tag{97c}$$

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

a.) *Mass flux:* $h\,\boldsymbol{v}\cdot\boldsymbol{n} = 0.2$;
b.) *Fixed height:* $h = 2$;
c.) *Velocity flux:* $\boldsymbol{v}\cdot\boldsymbol{n} = 0.1$.



(a) $T = 220$



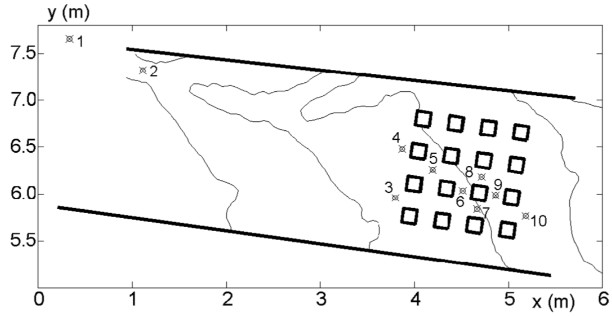
(b) $T = 300$

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

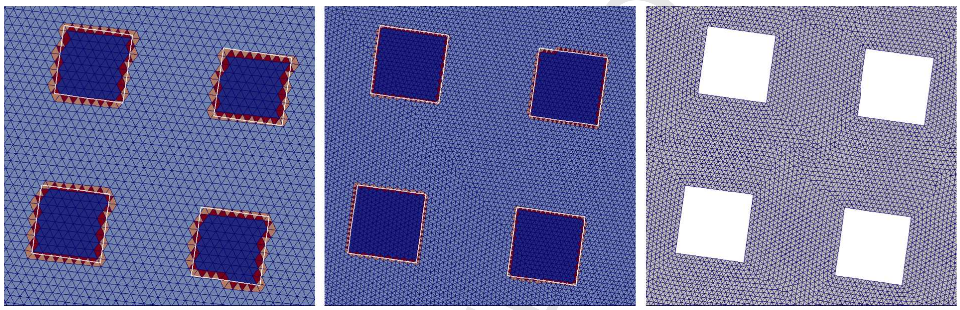
Penalties are introduced in the "fixed height" and "velocity flux" cases, with penalty set, as usual, as $\alpha = 2.0$. For the choices of fixed height and velocity flux boundary conditions, penalty terms are utilized with $\alpha = 2.0$. Results with all the previous three options are illustrated in Fig. 12. There is a discrepancy in the magnitude of errors between the conformal Nitsche and SB methods in the case of the mass flux and velocity flux boundary conditions, while for fixed height boundary condition this discrepancy is negligible. 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 Fig. 13. The geometry of this problem is considerably complicated and may require advanced meshing capability if a discretization using a conformal Nitsche method is pursued. This is 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 Fig. 14. 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



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



(b) Zoomed view on computational grids.

Fig. 18. Toce valley [53] flash flood test. Bathymetry and probe numbering and location are shown in Fig. 18(a). Zoomed views of the embedded and conformal grids near the buildings in the urban area are shown in Fig. 18(b): (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.

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 Fig. 15 and 16 at various times, while elevation plots of the height are presented in Fig. 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.

8.4. *Toce valley flash flood test: a simplified urban district layout*

In [53], 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 15*cm*, 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 Fig. 19. 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 of a zero initial water height, associated with the dry bed condition, a minimal threshold value $\epsilon = 1.0e-4$ is applied to the water height h, as suggested in [15]. Also, the characteristic element length in the discontinuity capturing term $(\partial_{x_i} \boldsymbol{W}\,,\, \nu\, \hat{h}_e^2 \hat{\boldsymbol{A}}_0\, \partial_{x_i} \boldsymbol{Y})_\Omega$ inside (51) is modified as,

$$\hat{h}_e = h_e + \sum_{i=1}^{n_{nd}^e} 0.5 \|\boldsymbol{d}\,(\boldsymbol{x}_i)\,\| \quad \text{if}\, \boldsymbol{x}_i \in \tilde{\Gamma}\,, \tag{98}$$

where n_{nd}^e is the number of nodes of element e that are on the surrogate boundary (none, one, two or three, for triangular elements). This strategy allows to adjust (increase) the dissipation at elements adjacent to the surrogate boundaries, de-

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

pending on the number of respective nodes on the surrogate boundary. This approach is of general applicability and robust, but the benefits were found to be most apparent in the case of this test. Also in this test, boundary penalty terms are added at building boundaries, for both the conformal and SB methods, with penalty parameter $\alpha = 2.0$.

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 Fig. 18, in the case of SB method, 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.
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 [53]. 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 Fig. 20 for the SB method simulation. Fig. 19 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 [30,43,48], 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.

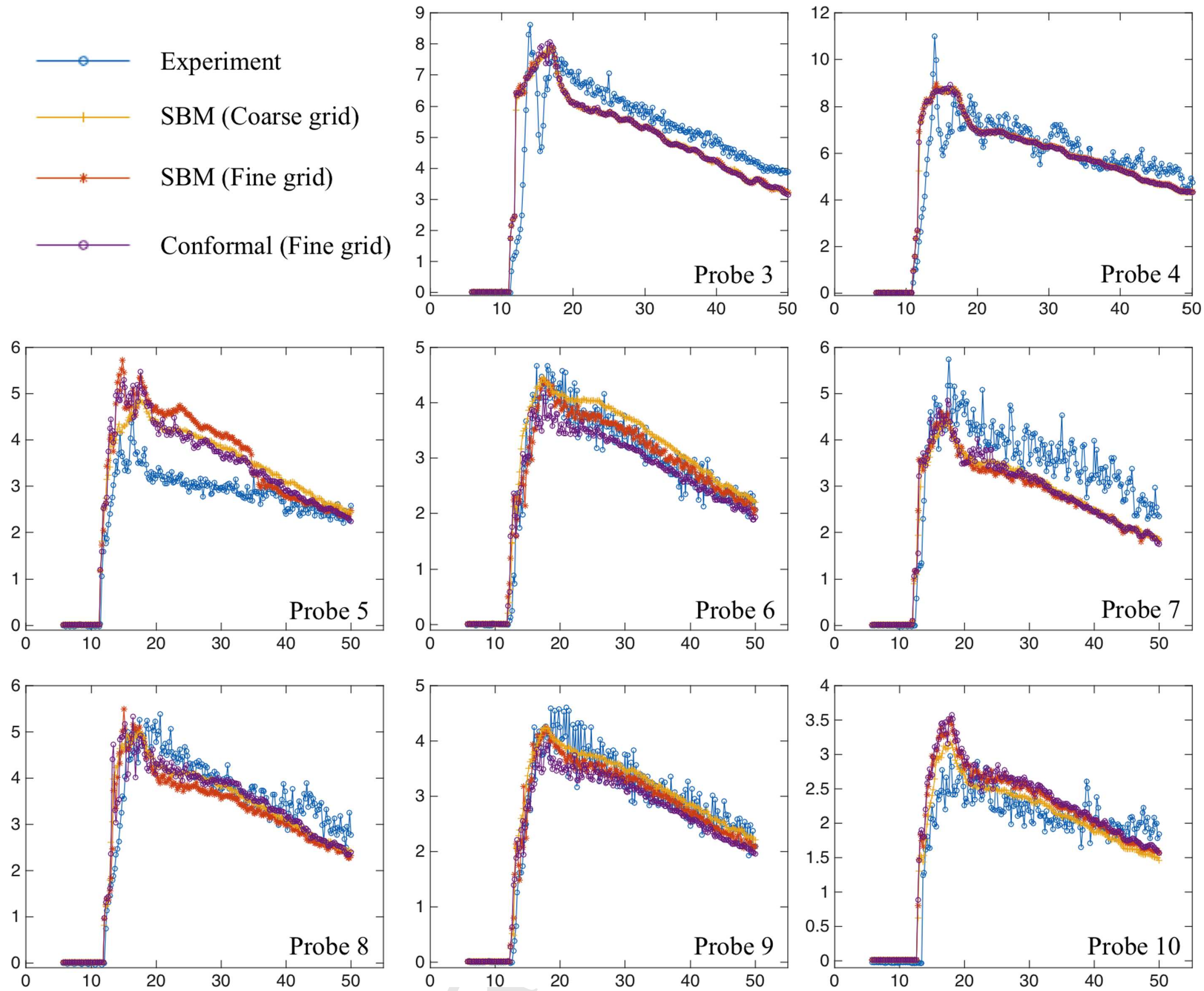
Acknowledgements

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, Smadar Karni, Computations of compressible multifluids, J. Comput. Phys. 169 (2) (2001) 594–623.
[2] Annavarapu Chandrasekhar, An Efficient Finite Element Method for Interface Problems, PhD thesis, Duke University, 2013.
[3] Chandrasekhar Annavarapu, Martin Hautefeuille, John E. Dolbow, A robust Nitsche's formulation for interface problems, Comput. Methods Appl. Mech. Eng. 225 (2012) 44–54.
[4] A. Bermudez, M.E. Vazquez, Upwind methods for hyperbolic conservation laws with source terms, Comput. Fluids 23 (8) (1994) 1049–1071.
[5] Daniele Boffi, Lucia Gastaldi, A finite element approach for the immersed boundary method, Comput. Struct. 81 (8) (2003) 491–501.
[6] P. Brufau, P. Garcia-Navarro, M.E. Vazquez-Cendon, Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography, Int. J. Numer. Methods Fluids 45 (2004) 1047–1082.
[7] Erik Burman, Ghost penalty, C. R. Math. 348 (21–22) (2010) 1217–1220.
[8] Erik Burman, Peter Hansbo, Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem, ESAIM Math. Model. Numer. Anal. 48 (3) (2014) 859–874.
[9] M.J. Castro, J. Gonzalez-Vida, C. Pares, Numerical treatment of wet/dry fronts in shallow flows with a roe scheme, Math. Models Methods Appl. Sci. 16 (6) (2006) 897–931.
[10] Derek M. Causon, David M. Ingram, Clive G. Mingham, A Cartesian cut cell method for shallow water flows with moving boundaries, Adv. Water Resour. 24 (8) (2001) 899–911.
[11] Derek M. Causon, David M. Ingram, Clive G. Mingham, G. Yang, Richard V. Pearson, Calculation of shallow water flows using a Cartesian cut cell approach, Adv. Water Resour. 23 (5) (2000) 545–562.
[12] Alexander Düster, Jamshid Parvizian, Zhengxiong Yang, Ernst Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Eng. 197 (45) (2008) 3768–3782.
[13] Charbel Farhat, Arthur Rallu, Sriram Shankaran, A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions, J. Comput. Phys. 227 (16) (2008) 7674–7700.
[14] Anita Hansbo, Peter Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Eng. 191 (47) (2002) 5537–5552.
[15] Guillermo Hauke, A symmetric formulation for computing transient shallow water flows, Comput. Methods Appl. Mech. Eng. 163 (1–4) (1998) 111–122.
[16] Guillermo Hauke, A stabilized finite element method for the Saint-Venant equations with application to irrigation, Int. J. Numer. Methods Fluids 38 (10) (2002) 963–984.

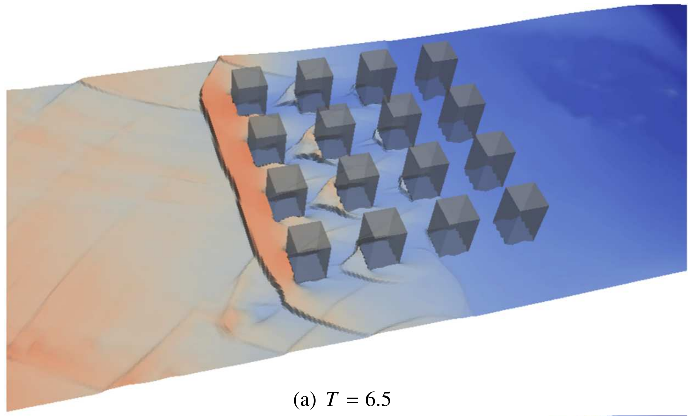
Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052



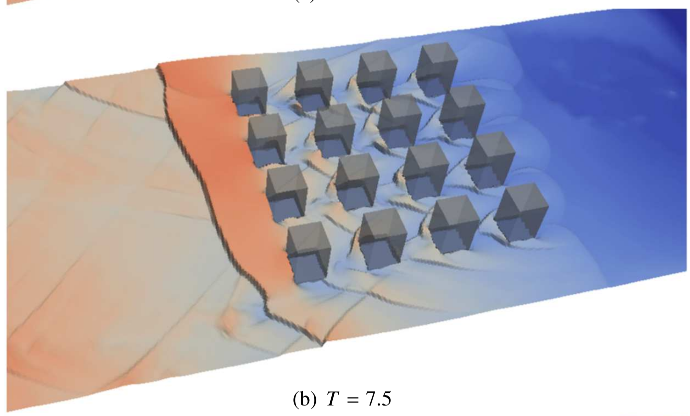
(a) Height of water at the probe locations 3 through 10, as depicted in Figure 18(a).

Fig. 19. Toce valley [53] flash flood test. Comparison of the numerical predictions for the water height with experimental data for probe locations 3 through 10. Compare with Fig. 18(a) for probe location. The horizontal axis represents time in seconds and the vertical axis represents water depth in centimeters.

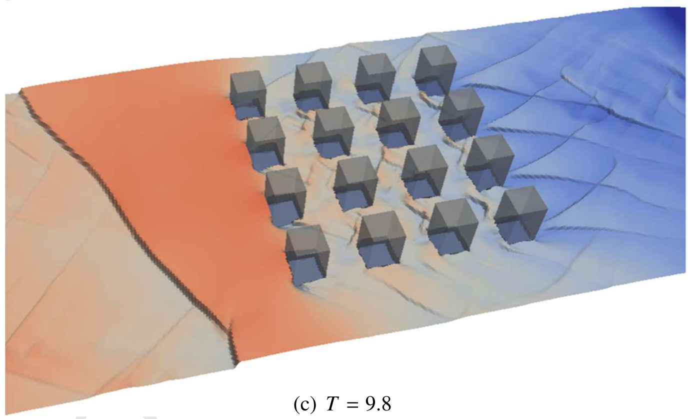
[17] Klaus Höllig, Finite Element Methods with B-Splines, SIAM, 2003.
[18] Klaus Höllig, Ulrich Reif, Joachim Wipper, Weighted extended b-spline approximation of Dirichlet problems, SIAM J. Numer. Anal. 39 (2) (2001) 442–462.
[19] K. Wang, A Computational Framework Based on an Embedded Boundary Method for Nonlinear Multi-Phase Fluid-Structure Interactions, PhD thesis, Stanford University, 2011.
[20] Y. Kagawa, T. Tsuchiya, T. Yamabuchi, H. Kawabe, T. Fujii, Finite-element simulation of nonlinear sound-wave propagation, J. Sound Vib. 154 (1) (1992) 125–145.
[21] Seongwon Kang, Gianluca Iaccarino, Frank Ham, Parviz Moin, Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method, J. Comput. Phys. 228 (18) (2009) 6753–6772.
[22] K. Kashiyama, H. Ito, M. Behr, T.E. Tezduyar, 3-step explicit finite-element computations of shallow-water flows on a massively-parallel computer, Int. J. Numer. Methods Fluids 21 (10) (1995) 885–900.
[23] K. Kashiyama, M. Kawahara, Selective lumping finite-element method for nearshore current, Int. J. Numer. Methods Fluids 4 (1) (1984) 71–97.
[24] K. Kashiyama, M. Kawahara, Boundary-type finite-element method for surface-wave motion based on trigonometric function interpolation, Int. J. Numer. Methods Eng. 21 (10) (1985).
[25] K. Kashiyama, M. Kawahara, A boundary-type finite-element model for water-surface wave problems, Int. J. Numer. Methods Fluids 8 (1) (1988) 65–79.
[26] K. Kashiyama, K. Saitoh, M. Behr, T.E. Tezduyar, Parallel finite element methods for large-scale computation of storm surges and tidal flows, Int. J. Numer. Methods Fluids 24 (12) (1997) 1371–1389.
[27] K. Kashiyama, M. Sakuraba, Adaptive boundary-type finite-element method for wave diffraction in harbors, Comput. Methods Appl. Mech. Eng. 112 (1–4) (1994) 185–197.
[28] M. Kawahara, H. Sakurai, K. Kashiyama, Boundary-type finite-element method for wave-propagation analysis, Int. J. Numer. Methods Fluids 8 (5) (1988) 559–578.
[29] M. Kawahara, N. Takeuchi, T. Yoshida, Two-step explicit finite-element method for tsunami wave-propagation analysis, Int. J. Numer. Methods Eng. 12 (2) (1978) 331–351.
[30] B. Kim, B.F. Sanders, J.E. Schubert, J.S. Famiglietti, Mesh type tradeoffs in 2d hydrodynamic modeling of flooding with a Godunov-based flow solver, Adv. Water Resour. 68 (2014) 42–61.



(a) $T = 6.5$



(b) $T = 7.5$



(c) $T = 9.8$

Fig. 20. Toce valley [53] flash flood test. Free surface elevation plots for the SB method solution.

[31] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems, J. Comput. Phys. (2017), https://doi.org/10.1016/j.jcp.2017.10.026, in press.
[32] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part II: linear advection–diffusion and incompressible Navier–Stokes equations, J. Comput. Phys. (2017), https://doi.org/10.1016/j.jcp.2018.01.023, in press.
[33] Hashem M. Mourad, John Dolbow, Isaac Harari, A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces, Int. J. Numer. Methods Eng. 69 (4) (2007) 772–793.
[34] J.A. Nitsche, Uber ein Variationsprinzip zur Losung Dirichlet-Problemen bei Verwendung von Teilraumen, die keinen Randbedingungen unteworfen sind, Abh. Math. Semin. Univ. Hamb. 36 (1971) 9–15.
[35] Jamshid Parvizian, Alexander Düster, Ernst Rank, Finite cell method, Comput. Mech. 41 (1) (2007) 121–133.
[36] Charles S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[37] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492.
[38] M. Ricchiuto, An explicit residual based approach for shallow water flows, J. Comput. Phys. 80 (2015) 306–344.
[39] M. Ricchiuto, A. Bollerman, Stabilized residual distribution for shallow water simulations, J. Comput. Phys. 228 (2009) 1071.
[40] Mario Ricchiuto, Andreas Bollermann, Stabilized residual distribution for shallow water simulations, J. Comput. Phys. 228 (4) (2009) 1071–1115.
[41] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, Comput. Methods Appl. Mech. Eng. 209 (2012) 266–283.

[42] T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid-structure interaction, Int. J. Numer. Methods Fluids 74 (9) (2014) 623–660.
[43] B.F. Sanders, J.E. Schubert, H.A. Gallegos, Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling, J. Hydrol. 362 (2008) 19–38.
[44] B. Schott, U. Rasthofer, V. Gravemeier, W.A. Wall, A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow, Int. J. Numer. Methods Eng. 104 (7) (2015) 721–748.
[45] B. Schott, W.A. Wall, A new face-oriented stabilized xfem approach for 2d and 3d incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 276 (2014) 233–265.
[46] G. Scovazzi, B. Carnes, Weak boundary conditions for wave propagation problems in confined domains: formulation and implementation using a variational multiscale method, Comput. Methods Appl. Mech. Eng. 221–222 (2012) 117–131.
[47] G. Scovazzi, T. Song, X. Zeng, A velocity/stress mixed stabilized nodal finite element for elastodynamics: analysis and computations with strongly and weakly enforced boundary conditions, Comput. Methods Appl. Mech. Eng. 325 (2017) 532–576.
[48] G. Vincent, S.F. Sandra, L. Julien, Z. Yves, Two-dimensional shallow-water model with porosity for urban flood modelling, J. Hydraul. Res. 46 (2008) 1.
[49] W.D. Smith, Application of finite-element analysis to body wave-propagation problems, Geophys. J. R. Astron. Soc. 42 (2) (1975) 747–768.
[50] T. Song, G. Scovazzi, A Nitsche method for wave propagation problems in time domain, Comput. Methods Appl. Mech. Eng. 293 (2015) 481–521.
[51] Eitan Tadmor, Skew-selfadjoint form for systems of conservation laws, J. Math. Anal. Appl. 103 (2) (1984) 428–442.
[52] S. Takase, K. Kashiyama, S. Tanaka, T.E. Tezduyar, Space–time SUPG formulation of the shallow-water equations, Int. J. Numer. Methods Fluids 64 (10–12) (2010) 1379–1394.
[53] G. Testa, D. Zuccala, F. Alcrudo, J. Mulet, S. Soares-Frazao, Flash flood flow experiment in a simplified urban district, J. Hydraul. Res. 45 (2007) 37–44.
[54] Kevin G. Wang, Patrick Lea, Alex Main, Owen McGarity, 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, American Society of Mechanical Engineers, 2014, pp. V08AT06A028–V08AT06A028.

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052

Sponsor names

Do not correct this page. Please mark corrections to sponsor names and grant numbers in the main text.

U.S. Department of Energy, *country*=United States, *grants*=SC0012169

Army Research Office, *country*=United States, *grants*=W911NF-13-1-0452

Highlights

- We propose a new embedded boundary method, in which the location and value of boundary conditions are shifted.
- This process yields a simple, robust, and efficient method, in which boundary conditions are enforced weakly.
- We apply the proposed method to hyperbolic systems: the equations of linear acoustic waves and shallow water flows.
- The benefit of the method is evident when considering the complex morphology of ocean coastlines or urban flooding scenarios.

Please cite this article in press as: T. Song et al., The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.04.052