Embedded domain Reduced Basis Models for the shallow water hyperbolic equations with the Shifted Boundary Method

Xianyi Zeng[a], Giovanni Stabile[b], Efthymios N. Karatzas[c], Guglielmo Scovazzi[d], Gianluigi Rozza[b]

[a]*Department of Mathematical Sciences, University of Texas at El Paso, United States.*
[b]*SISSA, International School for Advanced Studies, Mathematics Area, mathLab, Trieste, Italy.*
[c]*Department of Mathematics, School of Applied Mathematical and Physical Sciences, NTUA, Athens, Greece, and FORTH Institute of Applied and Computational Mathematics, Heraclion, Crete, Greece.*
[d]*Civil and Environmental Engineering Department, Duke University, Durham, NC 27708, United States.*

Abstract

We consider fully discrete embedded finite element approximations for a shallow water hyperbolic problem and its reduced-order model. Our approach is based on a fixed background mesh and an embedded reduced basis. The Shifted Boundary Method for spatial discretization is combined with an explicit predictor/multi-corrector time integration to integrate in time the numerical solutions to the shallow water equations, both for the full and reduced-order model. In order to improve the approximation of the solution manifold also for geometries that are untested during the offline stage, the snapshots have been pre-processed by means of an interpolation procedure that precedes the reduced basis computation. The methodology is tested on geometrically parametrized shapes with varying size and position.

1. Introduction

The computational cost associated with the numerical solution of partial differential equations might be in some cases prohibitive. This is happening, for example, when the numerical solution is required in nearly real time or a when large number of system configurations need to be tested. Shape optimization problems are a typical example of the latter case, where a large number of different geometrical configurations need to be analyzed to converge to an optimal solution. Reduced order models demonstrated to be a viable approach to reduce the computational burden and have been developed for a large variety of different linear and nonlinear problems [1, 2].

In recent times, immersed/embedded/unfitted methods have seen a great development from the seminal ideas of Peskin [3]. The key ideas in embedded methods is the use of grids that are not body-fitted, in which the geometry of the shapes to be simulated is immersed by way of computational geometry techniques. In this work, we base the reduced order models on the Shifted Boundary Method (SBM), which is an embedded/unfitted finite element method originally proposed for the Poisson, Stokes and incompressible Navier-Stokes equations [4, 5] and recently extended to wave equations and shallow water equations (SWE) [6]. In the SBM, a surrogate boundary is introduced in proximity of the true immersed boundary, and the boundary conditions are imposed on the surrogate boundary, with appropriate corrections that rely on Taylor expansions [4, 5]. The SBM does not require complicated data structures and numerical quadratures to integrate the governing equations on cut element, typical of cutFEM/XFEM approaches. Compared to other embedded finite element methods such as XFEM or cutFEM [7, 8, 9], the SBM has also the advantage that the degrees of freedom (unknowns) stay the same for varying geometries, hence existing reduced order model methodologies are more easily adapted. Specifically, the total number of unknowns in SBM is determined by the background mesh and it is independent of the location of the embedded geometry. In contrast, XFEM and other enriched finite element methods alike introduce new degrees of freedom such as the Heaviside functions within cut elements, hence the total number of unknowns typically varies with the embedded geometry locations and/or depends on a computationally expensive cutting of elements procedure.

Email addresses: xzeng@utep.edu (Xianyi Zeng), gstabile@sissa.it (Giovanni Stabile), karmakis@math.ntua.gr (Efthymios N. Karatzas), guglielmo.scovazzi@duke.edu (Guglielmo Scovazzi), grozza@sissa.it (Gianluigi Rozza)

In this article we focus our attention on projection-based reduced order models specifically tailored to geometrically parametrized problems [10, 11]. The idea is to combine the recently proposed Shifted Boundary Method [4, 5, 12] with Reduced Order Models based on the Proper Orthogonal Decomposition (POD) with Galerkin projection (SBM-ROM). This combination, that has been recently proposed in previous works in a different setting [13, 14, 15], allows to avoid the map of all the parametrized solutions to a common reference geometry, see also [16, 17]. In this paper, the embedded methodology introduced in the mentioned research works is extended to shallow water equations with explicit time marching schemes. The idea of merging embedded approaches with reduced order models has been proposed also in [18] where a fictitious domain method was coupled with a Proper Generalized Decomposition approach to study uncertain geometries. In [19] the authors proposed a projection based reduced order model starting from an embedded full order simulation applied to evolving interfaces.

With embedded simulations it is in fact easy to work with a common background mesh also in the case of large geometrical changes. By comparison, body-fitted meshes often require sophisticated re-meshing techniques, when complex geometrical deformations are present, and maintaining the topology of the underlying mesh is a difficult task.

In addition, we introduce a new approach to handle degrees of freedom located in the *"out of interest/ghost"* region which is based on a radial basis function interpolation. We denote this new approach as SBM-iROM. As shown in the numerical examples, this approach allows to partially reduce the drawback associated with the slow decay of the Kolmogorov N-width when dealing with embedded full order models.

Remark 1.1. Embedded computations with parametrized geometries are characterized by a slow decay of the Kolmogorov N-width which is caused by two different aspects. The first one is related to the embedded geometry that can arbitrarily move within the background mesh. Therefore, the boundary condition is applied on different elements, depending on the position of the true boundary with respect to the background mesh. The second aspect, that we believe is of minor entity, is related to the discontinuity of the solution in the proximity of the jump between the active and inactive nodes. The methodology that we have developed contributes to partially deal only with the second aspect.

The article is organized as follows: in section 2 we introduce the mathematical formulation of the full order problem, the associated weak formulation and the details concerning the specific discretization strategy. Section 3 describes in details the approach used for the construction of the reduced order model with a focus on the relevant changes required for the specific full order model formulation and introduces the employed interpolation preprocessing. In section 5 we introduce three numerical examples to show the properties and accuracy of the proposed methodology. Finally, in section 6 we report some conclusions and outlooks for future developments.

2. The model problem and the full order approximation

Before introducing the shallow water model, we briefly review the relevant literature: for linearized shallow water equations arising from the equations of acoustics we refer to the work [20], where only the generation of the second harmonic wave is considered (the higher order harmonics being neglected) under the assumption of weak non-linearity, while a set of uncoupled equations for the primary and secondary wave is discretized in space by a finite element method, and then solved by using the Newmark-β integration scheme for time. In [21], weak boundary conditions are considered for the hyperbolic structure of the wave equation based on stabilized methods and the variational multiscale analysis as well as we cite [22] for linear elastodynamics and [23] for Nitsche and wave propagation problems. In [24] one may see the shallow water equations as a symmetric advective-diffusive systems with source terms solved using SUPG and GLS stabilized methods via a predictor multi-corrector algorithm. The work of [25] treats two-dimensional shallow water equations applied to solve practical irrigation problems with large friction coefficients, dry bed conditions and singular infiltration terms. For a three-step shallow water flow explicit scheme, using parallel computing, and tidal flow in Tokyo Bay, we refer to [26], while a finite element method for the analysis of nearshore current, which is one of the principal currents in coastal seas analyzing two main characteristics of the wave, i.e. direction and height is introduced in [27]. Surface wave motion handled by the Helmholtz equation is studied in [28]. A new combinative method of boundary-type finite elements and boundary solutions to study wave diffraction-refraction and harbour oscillation problems is presented in [29] with model the mild-slope equation proved as an effective and accurate method for water surface wave problems. Large-scale computation of storm surges and tidal flows carried out with finite element methods are discussed in [30]. The work of [31] presents an adaptive boundary-type finite element method for wave diffraction-refraction in harbors model based on the mild-slope equation and an arbitrary reflection condition and in [32] boundary-type finite element method has been investigated and applied to the Helmholz and mild-slope equations useful for practical analysis. Numerical analysis of tsunamis applying the finite element method based on the shallow water wave equation with the Lax-Wendroff finite difference method used also for the analysis of the Tokachi-oki Earthquake tsunami problem and compared with the tide gauge records is investigated in [33]. A stabilized Residual Distribution scheme for the simulation of shallow water flows with a nonlinear variant of a Lax-Friedrichs type discretization is proposed in [34] while in [35] a numerical modeling of seismic body wave propagation problems is introduced. Finally, a space-time SUPG formulation of the shallow-water equations based on discontinuous Galerkin methods and residual redistribution schemes are examined in [36]. In particular, a complex morphology of the ocean coastlines in real scenarios is considered and complex coastlines as reflective walls in the framework of large-scale simulations of fine-scale urban are treated.

2.1. *Strong formulation of the shallow water problem*

The shallow-water equations, also known as de Saint-Venant equations [37], are a system of hyperbolic partial differential equations simulating the behavior of a free surface of a fluid [38, 39] when the depth of the fluid bed is shallow compared to the characteristic horizontal spatial length. Such system of equations is derived from the Navier-Stokes equations after integrating through the depth and observing that, since the horizontal length scale is much greater than the vertical length scale, the vertical component of the fluid velocity field is small compared to the horizontal component and the vertical gradients of pressure are nearly hydrostatic. This allows to conclude that the horizontal velocity field is constant over the entire depth of the fluid. Settings in fluid dynamics where the horizontal length scale is much greater than the vertical length scale are common and widely applicable, see e.g. atmospheric and oceanic modeling. As mentioned in the previous paragraph, the propagation of a tsunami wave can also be simulated efficiently with the shallow-water equations until it reaches the coast, see e.g. [6] and references therein. The shallow-water equations we examine are simulating thin layers of fluid in hydrostatic balance (with constant density), with upper and lower bounds: a free surface and the bed/ground topography, respectively. These systems can be formulated as

$$h_t + (hv_1)_x + (hv_2)_y = 0, \tag{2.1a}$$

$$(hv_1)_t + (hv_1^2 + \frac{1}{2}gh^2)_x + (hv_1v_2)_y = S_1, \tag{2.1b}$$

$$(hv_2)_t + (hv_1v_2)_x + (hv_2^2 + \frac{1}{2}gh^2)_y = S_2, \tag{2.1c}$$

where we indicate by h the height of the water, z the bathymetry of the water bed, and by $\eta = h + z$ the free surface measure, as it is visualized in Figure 2.1. The two-dimensional position and velocity vectors in Cartesian coordinate axes are denoted by $x = (x, y)$ and $v = (v_1, v_2)$. The quantity $\partial/\partial t$ expresses derivation with respect to time, and the source term $\mathbf{S} = (S_1, S_2) = gh(S_{o_1} - S_{f_1}, S_{o_2} - S_{f_2})$, for $S_{o_i} = -\partial z/\partial x_i$, $i = 1, 2$ and $S_{f_i} = f^2 v_i (v_1^2 + v_2^2)^{\frac{1}{2}} h^{-\frac{4}{3}}$ to be the slope of the river bed or ocean floor, and the friction (in terms of the Manning's roughness coefficient f) respectively. Further, g is the acceleration due to gravity. We note that other types of force can be added to the source term if needed, like wind stress, Coriolis forces, etc.

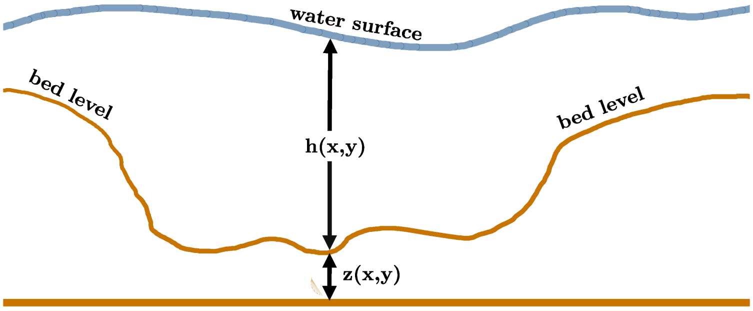


Figure 2.1: A sketch of the bathymetry $z(x, y)$ and the water height $h(x, y)$.

2.2. *Weak formulation on body-fitted grids*

The SWE is solved by a stabilized piecewise linear nodal finite element method (FEM) using two-dimensional triangular elements. We first introduce the notation and the weak formulation assuming a body-fitted grid and the extension to embedded boundaries using the Shifted Boundary Method is given in the next subsection. Let us denote the vector of conserved variables by $\boldsymbol{U} = [h,\ hv_1,\ hv_2]^T$ and the flux function by $\boldsymbol{F}(\boldsymbol{U})$ with components $\boldsymbol{F}_i,\ 1 \leq i \leq d = 2$; then the equation (2.1) can be written compactly as:

$$\partial_t \boldsymbol{U} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{Z}\ , \quad (\boldsymbol{x},\ t) \in \mathcal{D} \times [0,\ T]\ , \tag{2.2}$$

where $\boldsymbol{Z} = [0,\ S_1,\ S_2]^T$. Let $\mathcal{T}$ be a tessellation of the domain $\mathcal{D}$, we define the globally continuous piecewise linear solution and trial spaces:

$$\mathcal{S}^h = \mathcal{V}^h = \left\{ \boldsymbol{U}^h \in [C(\mathcal{D})]^{1+d}\ :\ \boldsymbol{U}^h\big|_K \in [\mathbb{P}_1(K)]^{1+d},\ \ \forall K \in \mathcal{T} \right\} \subset H^1(\mathcal{D}) \times H^{\text{div}}(\mathcal{D})\ , \tag{2.3}$$

where no boundary conditions are specified as they are enforced weakly. The semi-discrete stabilized finite element method states as finding $\boldsymbol{U}^h : [0,\ T] \mapsto \mathcal{S}^h$ such that for all $0 \leq t \leq T$ and $\boldsymbol{W}^h \in \mathcal{V}^h$:

$$(\boldsymbol{W}^h,\ \partial_t \boldsymbol{U}^h - \boldsymbol{Z})_{\mathcal{D}} - (\nabla \boldsymbol{W}^h,\ \boldsymbol{F})_{\mathcal{D}} + a_{\text{vms}}(\boldsymbol{W}^h,\ \boldsymbol{U}^h) + b(\boldsymbol{W}^h,\ \boldsymbol{U}^h) = 0\ . \tag{2.4}$$

Here $(\cdot,\ \cdot)_{\mathcal{D}}$ denotes the standard inner product for L^2 scalar, vectorial, and tensorial functions, a_{vms} contains a variational multiscale stabilization (VMS) term, and b contains all boundary terms.

The purpose of the VMS term is to prevent spurious oscillations due to the fact that equal-order interpolation is used for the velocity and fluid height variables. The spatial and full differential operators in linearized form are defined as:

$$\mathcal{L}\boldsymbol{U} = \sum_{i=1}^{d} \boldsymbol{A}_i \partial_{x_i} \boldsymbol{U}\ , \quad \mathcal{L}_t \boldsymbol{U} = \partial_t \boldsymbol{U} + \mathcal{L}\boldsymbol{U}\ , \tag{2.5}$$

where $\boldsymbol{A}_i \overset{\text{def}}{=\!=} \partial \boldsymbol{F}_i / \partial \boldsymbol{U}$. Then we may compute the dual operator of $\mathcal{L}$ as $\mathcal{L}^* = \sum_{i=1}^{d} \boldsymbol{A}_i^T \partial_{x_i}$ and define the VMS term as:

$$a_{\text{vms}}(\boldsymbol{W}^h,\ \boldsymbol{U}^h) = (\mathcal{L}^* \boldsymbol{W}^h,\ \tau_{\text{vms}} \boldsymbol{A}_0^{-1} (\mathcal{L}_t \boldsymbol{U}^h - \boldsymbol{Z}))_{\mathcal{D}}\ , \tag{2.6}$$

where τ_{vms} is a parameter that scales with time and it is computed as $\tau_{\text{vms}} = c_{\text{vms}} \mathit{\Delta} t/2$ with $\mathit{\Delta} t$ being the time step size and $c_{\text{vms}} = O(1)$ being a user defined parameter. A fixed value $c_{\text{vms}} = 2.0$ is used in all computations in this work. The matrix $\boldsymbol{A}_0$ is the Jacobian matrix converting the conservative variables $\boldsymbol{U}$ to primitive ones $\boldsymbol{Y} = [h,\ v_1,\ v_2]^T$:

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ v_1 & h & 0 \\ v_2 & 0 & h \end{bmatrix}\ . \tag{2.7}$$

Other options for the scaling matrix $\boldsymbol{A}_0^{-1}$ include the Jacobian matrix between conservative and entropy variables, as suggested in [40]; we adopt (2.7) for simplicity.

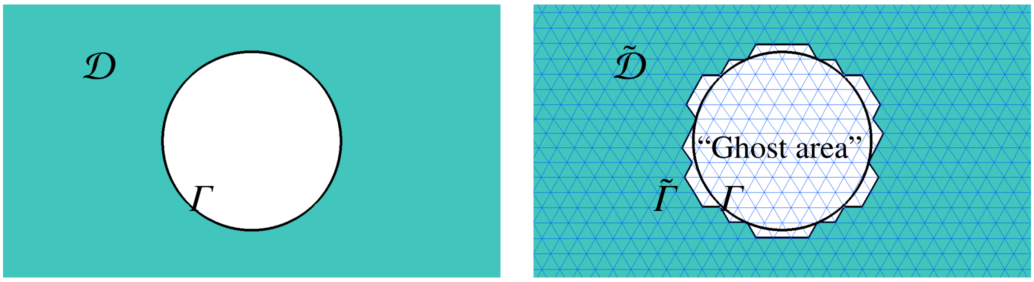
Remark. In principle, a discontinuity capturing (artificial viscosity) term could also be included for improved stability, especially when strong shocks are present. In this work, the problems considered do not involve strong shocks and we omit the discontinuity capturing operator. We point out however that the proposed methodology can be applied to the case in which shock capturing operators are used.

Lastly, the boundary condition can be classified into different types depending on the information available, such as incoming or outgoing flows, or subcritical or supercritical velocities. We shall only consider the boundary terms that are relevant to the test problem here, namely the Neumann condition, the subcritical inflow condition, and the subcritical outflow condition, defined on the boundary portions Γ_N, $\Gamma_{I;sub}$, and $\Gamma_{O;sub}$, respectively, see Figure 5.1; for a complete list of all boundary conditions the readers are referred to [41]. Particularly, the three boundary conditions are given by:

$$\boldsymbol{v} \cdot \boldsymbol{n} = v_N\ , \qquad \boldsymbol{x} \in \Gamma_N\ , \tag{2.8}$$

$$h\boldsymbol{v} \cdot \boldsymbol{n} = m_{I;sub}\ , \quad \boldsymbol{v} \cdot \boldsymbol{\tau} = 0\ , \qquad \boldsymbol{x} \in \Gamma_{I;sub}, \tag{2.9}$$

$$h\boldsymbol{v} \cdot \boldsymbol{n} = m_{O;sub}\ , \qquad \boldsymbol{x} \in \Gamma_{O;sub}\ . \tag{2.10}$$



(a) The geometry $\mathcal{D}$ surrounding a disk and its boundary Γ.

(b) The SBM surrogate geometry $\tilde{\mathcal{D}}$, the surrogate boundary $\tilde{\Gamma}$, and ghost area.

Figure 2.2: Embedded geometry: (A) The geometry of a disk and (B) the SBM surrogate geometry attached with the ghost area.

Here $\boldsymbol{n}$ is the outer unit normal on the boundary, $\boldsymbol{\tau}$ is the tangent vector, and v_N, $m_{I;sub} < 0$, and $m_{O;sub} > 0$ are prescribed normal velocity, inflow mass rate, and outflow mass rate, respectively. Note that in principle (2.9) and (2.10) are only valid when the Froude number is smaller than unity, otherwise the boundary condition needs to be switched to supercritical ones; the latter scenario, however, does not occur in the tests in this work. While we shall describe the Neumann condition by assuming a general v_N, in all cases the value of v_N is set to zero so that the boundary either represent a slippery wall or a symmetry plane. To this end, the boundary term b is given by:

$$b(\boldsymbol{W}^h,\ \boldsymbol{U}^n) = \langle \boldsymbol{W}^h, \boldsymbol{H}^h_N \rangle_{\Gamma_N} + \langle \boldsymbol{W}^h, \boldsymbol{H}^h_{I;sub} \rangle_{\Gamma_{I;sub}} + \langle \boldsymbol{W}^h, \boldsymbol{H}^h_{O;sub} \rangle_{\Gamma_{O;sub}} \ , \tag{2.11}$$

where the angle brackets are the inner product on a general boundary piece Γ that is defined as $\langle \boldsymbol{W}, \boldsymbol{H} \rangle_\Gamma = \int_\Gamma \boldsymbol{W} \cdot \boldsymbol{H} d\Gamma$. Particularly, the vectors $\boldsymbol{H}^h_N$, $\boldsymbol{H}^h_{I;sub}$, and $\boldsymbol{H}^h_{O;sub}$ are given respectively by:

$$\boldsymbol{H}^h_N = v_N \begin{bmatrix} h \\ h\boldsymbol{v} \end{bmatrix} + \frac{1}{2} g h^2 \begin{bmatrix} 0 \\ \boldsymbol{n} \end{bmatrix} \ , \quad \boldsymbol{H}^h_{I;sub} = m_{I;sub} \begin{bmatrix} 1 \\ (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n} \end{bmatrix} + \frac{1}{2} g h^2 \begin{bmatrix} 0 \\ \boldsymbol{n} \end{bmatrix} \ , \quad \boldsymbol{H}^h_{O;sub} = m_{O;sub} \begin{bmatrix} 1 \\ \boldsymbol{v} \end{bmatrix} + \frac{1}{2} g h^2 \begin{bmatrix} 0 \\ \boldsymbol{n} \end{bmatrix} \ . \tag{2.12}$$

In the case of an embedded boundary, the boundary term (especially the Neumann one) needs to be modified, as described next.

2.3. Discretization: the Shifted Boundary Method

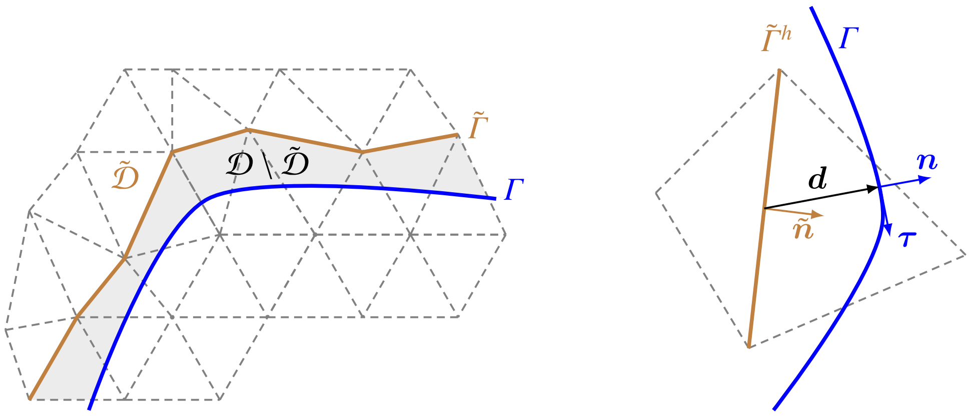
In this subsection, we introduce the basic aspects of the Shifted Boundary Method adapted to the shallow water equations [4, 5, 23, 6]. We consider a surrogate domain $\tilde{\mathcal{D}}$ and boundary $\tilde{\Gamma}$ together with the true computational domain $\mathcal{D}$ and its boundary Γ, as shown in Figure 2.2 and Figure 2.3a. We indicate by $\tilde{\boldsymbol{n}}$ the unit outward-pointing normal to the surrogate boundary $\tilde{\Gamma}$, which is distinct from the outward-pointing normal $\boldsymbol{n}$ to Γ as seen in Figure 2.3b. $\tilde{\Gamma}$ consists of the edges that are closest in some sense to the true boundary Γ, as shown in Figure 2.3b. The mapping

$$\boldsymbol{M}^h : \tilde{\Gamma} \to \Gamma, \tag{2.13}$$

is introduced, similarly to [12, Section 2.1], which maps any point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}$ on the surrogate boundary, to a point $\boldsymbol{x} = \boldsymbol{M}^h(\tilde{\boldsymbol{x}})$ on the true physical boundary Γ. Through $\boldsymbol{M}^h$, an auxiliary distance vector function $\boldsymbol{d}_{\boldsymbol{M}^h}$ is defined as

$$\boldsymbol{d}_{\boldsymbol{M}^h}(\tilde{\boldsymbol{x}}) \ = \ \boldsymbol{x} - \tilde{\boldsymbol{x}} \ = \ [\, \boldsymbol{M}^h - \boldsymbol{I} \,](\tilde{\boldsymbol{x}}) \ . \tag{2.14}$$

For brevity, we set $\boldsymbol{d} = \boldsymbol{d}_{\boldsymbol{M}^h}$ and write $\boldsymbol{d} = \|\boldsymbol{d}\|\boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is the unit vector in the direction of $\boldsymbol{d}$. In the case of smooth surfaces with one type of boundary condition, the mapping $\boldsymbol{M}^h$ corresponds to the closest point projection and $\boldsymbol{\nu} = \boldsymbol{n}$, see e.g. Figure 2.3b. The general construction and analysis of $\boldsymbol{M}^h$ are detailed in [12], including the case when corners are present or the closures of the Dirichlet boundary Γ_D and the Neumann boundary Γ_N have non-empty intersections. Following the analysis therein, we may hypothesize that $\boldsymbol{M}^h$ is continuous and Lipschitz. This assumption makes sense since the true surface is smooth between edges and corners. The mapping $\boldsymbol{M}^h$ can be used to extend the unit normal vector $\boldsymbol{n}$ from the boundary Γ to the surrogate boundary $\tilde{\Gamma}$ as $\bar{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) \equiv \boldsymbol{n}(\boldsymbol{M}^h(\tilde{\boldsymbol{x}}))$. In the next sections, we use the short-hand notation $\boldsymbol{n}(\tilde{\boldsymbol{x}})$, which means $\bar{\boldsymbol{n}}(\tilde{\boldsymbol{x}})$ at a point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}$. In a similar way, we extend the boundary conditions on Γ to the boundary $\tilde{\Gamma}$ of the surrogate domain.



(a) The true geometry $\mathcal{D}$, the surrogate geometry $\tilde{\mathcal{D}}$ and the corresponding boundaries $\tilde{\Gamma}$ and Γ.

(b) The auxiliary distance vector $\boldsymbol{d}$, the true normal and tangent vector $\boldsymbol{n}$ and $\boldsymbol{\tau}$ respectively.

Figure 2.3: The true and the surrogate geometry and boundary, and the SBM related quantities $\boldsymbol{d}$, τ, $\boldsymbol{n}$, $\tilde{\boldsymbol{n}}$.

2.3.1. *Semi-discrete shifted boundary weak formulation*

We can now set up the semi-discrete SBM weak formulation relying on the surrogate domain $\tilde{\mathcal{D}}$ whose boundary is composed of both body-fitted and embedded portions. With a slight abuse of notation, we indicate with Γ the portion of the boundary $\partial\tilde{\mathcal{D}}$ that is body-fitted, and by $\tilde{\Gamma}$ the portion of the boundary $\partial\tilde{\mathcal{D}}$ that is embedded (see, e.g., Figure 5.1, where Γ consists of the four exterior edges and $\tilde{\Gamma}$ is the surrogate boundary associated with the internal circle). We discretize now $\tilde{\mathcal{D}}$ using a mesh triangulation $\tilde{\mathcal{D}}_{\mathcal{T}}$ consisting of triangles K that belong to a tessellation $\mathcal{T}$. The weak SBM formulation is now given as:

Find $\boldsymbol{U}^h : [0,\ T] \mapsto \tilde{\mathcal{S}}^h$, such that for all $0 \le t \le T$ and $\boldsymbol{W}^h \in \tilde{\mathcal{V}}^h$:

$$(\boldsymbol{W}^h,\ \partial_t \boldsymbol{U}^h - \boldsymbol{Z})_{\tilde{\mathcal{D}}} - (\nabla \boldsymbol{W}^h,\ \boldsymbol{F})_{\tilde{\mathcal{D}}} + a_{\text{vms}}(\boldsymbol{W}^h,\ \boldsymbol{U}^h) + b(\boldsymbol{W}^h,\ \boldsymbol{U}^h) + \tilde{b}(\boldsymbol{W}^h,\ \boldsymbol{U}^h) = 0\ , \tag{2.15}$$

where $\tilde{\mathcal{S}}^h$ and $\tilde{\mathcal{V}}^h$ are obtained by replacing $\mathcal{D}$ in (2.3) with the surrogate $\tilde{\mathcal{D}}$. The VMS term remains the same as (2.6), except the inner product is evaluated on $\tilde{\mathcal{D}}$; the boundary term b is given by (2.11) for all body-fitted boundaries; and the second boundary term $\tilde{b}$ is given below in the case of an embedded Neumann boundary ($\tilde{\Gamma} = \tilde{\Gamma}_N$):

$$\tilde{b}(\boldsymbol{W}^h,\ \boldsymbol{U}^h) = \langle \boldsymbol{W}^h,\ \tilde{\boldsymbol{H}}^h_N \rangle_{\tilde{\Gamma}_N}\ , \quad \tilde{\boldsymbol{H}}^h_N = \Big((v_N - \boldsymbol{n}^T \nabla \boldsymbol{v} \boldsymbol{d}) \boldsymbol{n} \cdot \tilde{\boldsymbol{n}} + (\boldsymbol{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau} \cdot \tilde{\boldsymbol{n}} \Big) \begin{bmatrix} h \\ h\boldsymbol{v} \end{bmatrix} + \frac{1}{2} g h^2 \begin{bmatrix} 0 \\ \tilde{\boldsymbol{n}} \end{bmatrix}\ . \tag{2.16}$$

The first term in the definition of $\tilde{\boldsymbol{H}}^h_N$ attempts to enforce a normal velocity *shifted* to the surrogate interface by using Taylor series expansions:

$$\begin{aligned}
\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \tilde{\boldsymbol{n}}(\tilde{\boldsymbol{x}}) &= [(\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{n})\boldsymbol{n} + (\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{\tau})\boldsymbol{\tau}] \cdot \tilde{\boldsymbol{n}} \\
&\approx \{[(\boldsymbol{v}(\boldsymbol{x}) - \nabla \boldsymbol{v}(\tilde{\boldsymbol{x}})\boldsymbol{d})\boldsymbol{n}] \cdot \boldsymbol{n} + (\boldsymbol{v}(\tilde{\boldsymbol{x}}) \cdot \boldsymbol{\tau})\boldsymbol{\tau}\} \cdot \tilde{\boldsymbol{n}} = (v_N - \boldsymbol{n}^T \nabla \boldsymbol{v} \boldsymbol{d})\boldsymbol{n} \cdot \tilde{\boldsymbol{n}} + (\boldsymbol{v} \cdot \boldsymbol{\tau})\boldsymbol{\tau} \cdot \tilde{\boldsymbol{n}}\ .
\end{aligned}$$

Embedded boundary conditions of other types can be derived similarly but are omitted here, for the sake of brevity, since they are not applied in the test problems considered here.

2.3.2. *Time discretization*

An explicit predictor/multi-corrector (PMC) time integration is used to march the numerical solutions in time [23, 6]. To this end, let us denote the time ordinate by t and the spatial coordinate by $\boldsymbol{x}$. Furthermore, the superscript ^{n} designates a variable associated with the n^{th} time step t^n and a subscript $_A$ designates a variable associated with a mesh node $\boldsymbol{x}_A$, $1 \le A \le n_{\text{node}}$, where n_{node} is the total number of nodes.

The discrete solution vector at t^n is given by:

$$\boldsymbol{U}(t^n, \boldsymbol{x}) \approx \boldsymbol{U}^n(\boldsymbol{x}) \overset{\text{def}}{=} \sum_{A=1}^{n_{\text{node}}} \boldsymbol{U}_A^n N_A(\boldsymbol{x}) \;, \tag{2.17}$$

where $N_A(\boldsymbol{x})$ is the piecewise linear shape function associated with the node $\boldsymbol{x}_A$ and $\boldsymbol{U}_A^n$ are the degrees of freedom at the same node. Choosing $\boldsymbol{W}^h = \boldsymbol{e}_i N_A(\boldsymbol{x})$ for all $1 \leq A \leq n_{\text{node}}$ and $1 \leq i \leq 1+d$ in the weak formulation (2.4), where $\boldsymbol{e}_i$ is the i^{th} unit vector in $\mathbb{R}^{1+d}$, we obtain a system of ordinary differential equations:

$$\mathbf{M}\dot{\mathbf{U}}(t) + \mathcal{R}(\mathbf{U}) = 0 \;, \tag{2.18}$$

where $\mathbf{U}(t) : \mathbb{R}^+ \mapsto \mathbb{R}^{(1+d)n_{\text{node}}}$ contains all degrees of freedom at all nodes, i.e., $\boldsymbol{U}_A \in \mathbb{R}^{1+d}$ for all $1 \leq A \leq n_{\text{node}}$ and $\mathbf{M}$ is the diagonal lumped mass matrix. The operator $\mathcal{R} : \mathbb{R}^{(1+d)n_{\text{node}}} \mapsto \mathbb{R}^{(1+d)n_{\text{node}}}$ contains the (spatial) residual at each node for each component of the solution variable and includes the standard finite element terms in continuous Galerkin formulation, the stabilization term, and the Shifted Boundary Method terms that arise in the transmission boundary condition at the surrogate boundary.

To update the solution vector from $\mathbf{U}^n$ to $\mathbf{U}^{n+1}$, the explicit PMC method can be considered as a fixed-point iteration approximation for the midpoint rule,

$$\mathbf{M}(\mathbf{U}^{n+1} - \mathbf{U}^n) + \delta t^n \mathcal{R}((\mathbf{U}^n + \mathbf{U}^{n+1})/2) = 0 \;, \tag{2.19}$$

where $\delta t^n = t^{n+1} - t^n > 0$ is the time step size. In PMC, one seeks successive approximations to $\mathbf{U}^{n+1}$, denoted by $\mathbf{U}^{(k)}$ where $k = 0, 1, \cdots$. In particular $\mathbf{U}^{(0)} \overset{\text{def}}{=} \mathbf{U}^n$ is the "predictor" of $\mathbf{U}^{n+1}$ and once $\mathbf{U}^{(k)}$, $k \geq 0$ is computed, one computes the next iterate $\mathbf{U}^{(k+1)}$ by:

$$\mathbf{M}(\mathbf{U}^{(k+1)} - \mathbf{U}^n) + \delta t^n \mathcal{R}((\mathbf{U}^n + \mathbf{U}^{(k)})/2) = 0 \;. \tag{2.20}$$

In practice, one terminates the iteration after a preset number of correctors (each $\mathbf{U}^{(k)}$ with $k \geq 1$ is known as a "corrector"), i.e., (2.20) is performed for $k = 0, \cdots, n_{\text{pmc}} - 1$ where typical values for n_{pmc} is between 2 and 4. Finally, the update $\mathbf{U}^{n+1} = \mathbf{U}^{(n_{\text{pmc}})}$ is applied.

Remark 2.1. If one sets $n_{\text{pmc}} = 1$, the method is equivalent to the explicit forward-Euler method, whereas $n_{\text{pmc}} = 2$ gives an implementation of the second-order Runge-Kutta scheme.

3. Reduced order model with a POD-Galerkin method

The reduced order model proposed here is based on a POD-Galerkin approach. It means that the underlying system of equations is projected onto a linear subspace of smaller dimension spanned by a reduced number of global basis functions (POD modes). There are different techniques to generate this linear subspace and here we rely on the POD [42]. The overall methodology is based on the classic offline-online splitting approach [2], which is briefly recalled in what follows.

Offline Stage

During the offline stage we start with a parametric partial differential equation which is parametrized by means of a p-dimensional parameter vector $\boldsymbol{\mu} \in \mathcal{P}$. The full order model is then solved for a finite dimensional set of training points in $\{\boldsymbol{\mu}_i\}_{i=1}^{N_{\mu_{\text{train}}}} \subset \mathcal{P}$.

In the current framework we construct one linear subspace which includes both parameter and time variations. This means that the snapshots matrix on which the POD is based is assembled as:

$$S_U = [U(t_1, \boldsymbol{\mu}_1), U(t_2, \boldsymbol{\mu}_1), \ldots, U(t_{N_t}, \boldsymbol{\mu}_{N_{\mu_{\text{train}}}}))] \in \mathbb{R}^{N_h \times N_s}, \tag{3.1}$$

where $N_h = (1+d)n_{\text{node}}$ is the dimension of the FOM solution vector, $N_s = N_t \cdot N_{\mu_{\text{train}}}$ is the total number of stored snapshots where N_t is the number of time steps in which we store the solution and $N_{\mu_{\text{train}}}$ is the number of training samples in the parameter space. The POD modes are generated using the method of snapshots as originally proposed by Sirovich [42] which relies on the solution of an eigenproblem on the correlation matrix:

$$C = S_U^T S_U, \tag{3.2}$$

and on the computation of the POD bases exploiting the resulting eigenvalues $\{\lambda_i\}_{i=1}^{N_s}$ ad eigenvectors $\{\psi_i\}_{i=1}^{N_s}$:

$$\phi_i = \frac{1}{\sqrt{\lambda_i}} S_U \psi_i. \tag{3.3}$$

This operation results into to the POD space:

$$\boldsymbol{\Phi}_{\text{full}} = \text{span}(\phi_1, \ldots, \phi_{N_s}). \tag{3.4}$$

Remark 3.1. Based on the eigenvalue decomposition of the correlation matrix C it is possible to discard some of the modes and to create a POD space $\boldsymbol{\Phi}_{\text{r}}$ that includes a limited number of the computed POD modes. This space will be employed for the subsequent Galerkin projection. The correlation matrix has been computed relying onto the Frobenius inner product. Other options are possible (such as L^2 or H^1 norms) but in the current setting, due to the fact that the underlying background mesh is made of finite elements with a similar size, for the sake of simplicity, we have decided to rely on the Frobenius norm to calculate the POD modes.

3.1. Snapshots Interpolation, the SBM-iROM

The procedure described above is rather straightforward for body-fitted meshes. For unfitted meshes there are additional complexities that need to be addressed. One of them is related to the inactive nodes that belong to the so-called *ghost area*. These nodes are embedded into the body and do not play a role in the computation. As depicted in Figure 3.1, the number and location of the inactive nodes is changing depending on the shape of the parametrized geometry. In order to create a global basis function that can be used for any new parameter configuration it is necessary to handle also the inactive nodes. A possible approach would be to set them to a constant value. However, this approach would introduce a discontinuity in the solution field on the jump between active and inactive nodes. Therefore, we have decided to preprocess each snapshot with an interpolation strategy, in order to avoid discontinuity caused by deactivated elements. Another possible option to avoid this issue would be to compute an harmonic extension from the boundary to the deactivated nodes [43]. The method proposed here has the advantage of ensuring a sufficient level of smoothness without the solve of the additional partial differential equation problem required by the harmonic extension. The interpolant function has been evaluated for each snapshot considering only the active nodes and the values in the inactive nodes have been evaluated using the interpolant function. Each solution field has been replaced by:

$$U(\boldsymbol{x}) \rightarrow U_I(\boldsymbol{x}) = \sum_{k=1}^{N_{\text{active}}} \omega_k \varphi(\|\boldsymbol{x} - \boldsymbol{x}_k\|) + \sum_{j=1}^{m} b_j p_j(\boldsymbol{x}), \tag{3.5}$$

where ω_k are weights that needs to be determined imposing the interpolation condition, $\varphi(\|\boldsymbol{x} - \boldsymbol{x}_k\|)$ are radial basis functions where $\boldsymbol{x}_k$ are the coordinates of the active nodes, $\{p_i(\boldsymbol{x})\}_{i=1}^{m}$ are monomials that span the space of polynomials with a specific degree. Adding polynomials to the RBF interpolant functions helps to properly capture constant and linear features in the given data and ensures positive-definiteness of the RBF function, which in turn implies solvability of the interpolation problem [44]. The coefficient vectors $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_{N_{\text{active}}}]^T$ and $\boldsymbol{b} = [b_1, \ldots, b_m]^T$ are obtained solving the following linear system of equations:

$$\begin{cases} \left(\boldsymbol{K} + \sigma^2 \boldsymbol{I}\right)\boldsymbol{\omega} + \boldsymbol{P}\boldsymbol{b} = \boldsymbol{d}, \\ \boldsymbol{P}^T \boldsymbol{\omega} = \boldsymbol{0}, \end{cases} \tag{3.6}$$

where $(\boldsymbol{K})_{ij} = \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$, $(\boldsymbol{P})_{ij} = p_j(\boldsymbol{x}_i)$, σ is a smoothing parameter that eventually relaxes the interpolation condition in order to smoothen the "interpolant" function, which on this occasion has been chosen $\sigma = 0$. Selecting a RBF which is positive definite of order q with $\boldsymbol{P}$ that has full column rank, the solution is unique provided that the degree of the monomial terms is at least $m = q - 1$ [45, 44]. In the specific case, for each snapshot and solution field, the interpolant function has been constructed using observation only on the active nodes. The solution field on the inactive nodes has been reconstructed using the previously defined interpolant function:

$$U(\boldsymbol{x}_{i_{\text{inactive}}}) = U_I(\boldsymbol{x}_{i_{\text{inactive}}}). \tag{3.7}$$

Here we decided to rely on second-order polyharmonic splines that have the expression:

$$\varphi = (\epsilon r)^2 \log(\epsilon r), \tag{3.8}$$

where $r = \|\boldsymbol{x} - \boldsymbol{c}\|$ with $\boldsymbol{x}$ and $\boldsymbol{c}$ the evaluation points and the radial basis center respectively. ϵ is a scalar shape parameter that in this case is set $\epsilon = 1.0$. This function is conditionally positive definite provided that the order of the monomial is at least equal to 2 [45, 44]. Therefore, in order to ensure the uniqueness of the solution we have chosen to use $m = 2$ in expression (3.5). Using this setting, prior to the POD basis computation, the solution snapshots are replaced by the corresponding interpolant function. This operation permits to have a continuous solution field. We denote this model as SBM-iROM (Shifted Boundary Method with interpolation Reduced Order Model). In Figure 3.2 it is shown, for the height of the water field, a snapshot before and after the interpolation procedure. The hole on the left part of the picture corresponds to the inactive nodes which belongs to the embedded geometry and are set to zero by the full order model solver.

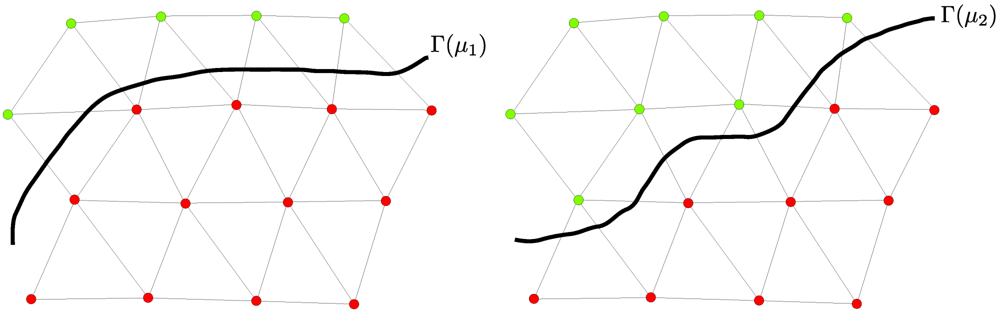


Figure 3.1: Visualization of active (green nodes) and inactive nodes (red nodes) for two different parametrized geometries

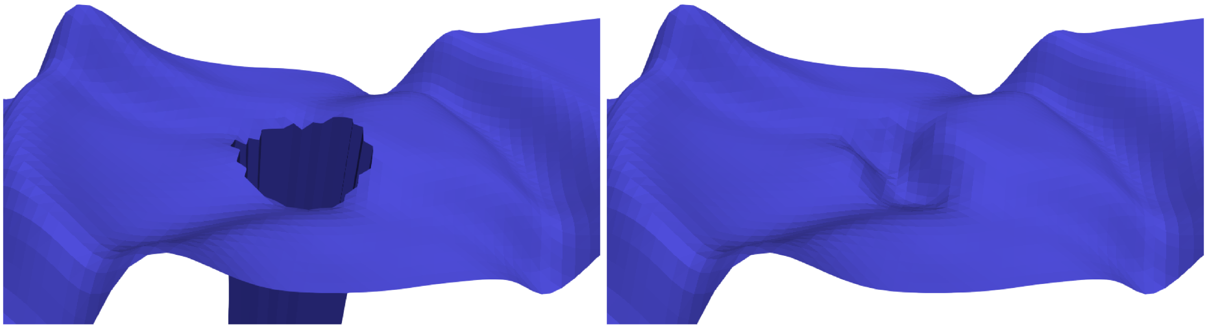


Figure 3.2: Visualization of a solution snapshot for the height of the water field, without interpolation (left image) and with interpolation (right image) procedure

3.2. *POD-Galerkin Projection*

Once the POD bases are computed one can use them to approximate the solution field $\boldsymbol{U} \in \mathbb{R}^{N_h}$ with its low dimensional representation $\boldsymbol{U}_r$ which is defined as a linear combination of N_r global basis functions $\boldsymbol{\Phi}_r = \text{span}(\phi_1, \dots, \phi_{N_r})$:

$$\boldsymbol{U} \approx \boldsymbol{U}_r = \boldsymbol{\Phi}_r \boldsymbol{a}, \tag{3.9}$$

Table 1: Sample computational costs associated with the residual term and the rest of (3.11) for Test 1, Case $R = 0.08$ in Section 5.1. The grid for this test contains 5,419 vertices and a total of 561 snapshots are used to construct the POD basis; note that only 29 POD modes (in bold fonts) are needed to produce ROM solution that agrees extremely well with the FOM one, in which case the residual computation takes 94% of the overall computational cost. All computations are performed in serial using a Linux machine with the Xeon E5026 2.10GHz CPU.

Number of POD Modes.	15	**29**	52	71	85
Cost on computing $\mathcal{R}()$ (sec).	2017	**2003**	1999	2565	2587
Cost on the rest of (3.11) (sec).	37	**136**	454	1125	1640

where the basis functions $\phi_i \in \mathbb{R}^{N_h}$ are computed by POD applied on the interpolated snapshots. The coefficients of the POD expansion are then retrieved by means of Galerkin projection of the original system of equations onto the space spanned by the POD modes. Also at the reduced order level we have decided to use exactly the same time marching as the one employed at the full order level. This results in:

$$\boldsymbol{\Phi}_r^T \mathbf{M} \boldsymbol{\Phi}_r (\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^n) + \delta t^n \boldsymbol{\Phi}_r^T \mathcal{R}(\boldsymbol{\Phi}_r (\boldsymbol{a}^n + \boldsymbol{a}^{(k)})/2) = 0. \tag{3.10}$$

That can be reformulated as:

$$\mathbf{M}_r (\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^n) + \delta t^n \boldsymbol{\Phi}_r^T \mathcal{R}(\boldsymbol{\Phi}_r (\boldsymbol{a}^n + \boldsymbol{a}^{(k)})/2) = 0. \tag{3.11}$$

In the expression above, $\mathbf{M}_r = \boldsymbol{\Phi}_r^T \mathbf{M} \boldsymbol{\Phi}_r \in \mathbb{R}^{N_r \times N_r}$ can be precomputed and does not depend on the input parameters $\boldsymbol{\mu}$. The problem can be therefore expressed in terms of reduced coefficients $\boldsymbol{a}$ with the only difficulty that we will have to assemble the residual $\mathcal{R}$ also at every iteration of the ROM problem. Moreover, the residual term $\mathcal{R}$ has a nonlinear dependency with respect to the input parameter $\boldsymbol{\mu}$ which parametrizes the embedded geometry. Such possibly expensive residual computation could be substituted by point-wise evaluation of the residual function in some selected points of the domain using an hyper-reduction technique such as the empirical interpolation method [46], the discrete empirical interpolation method [10, 47] or the Gauss–Newton with approximated tensors (GNAT) [48]. Further, for a discrete empirical method adapted to embedded methods we refer to [17]. However, in this work, since the main concern is to test the applicability of the methodology and the application of the proposed interpolation strategy, we have decided to assemble the full residual and to project it onto the reduced basis spaces at each iteration. In fact, we observe that the computation time spent on the full residual term $\mathcal{R}(\boldsymbol{\Phi}_r(\boldsymbol{a}^n + \boldsymbol{a}^{(k)})/2)$ is significant larger than that spent on the rest of the algorithm (3.11) for a reasonable number of POD modes, see Table 1. Thus we expect significant computation time reduction in ROM computations with an appropriate choice of hyper-reduction technique for the residual term. In what follows, for the sake of completeness, we provide a possible approach to deal with hyper-reduction of the method.

4. A possible implementation of hyper-reduction using the discrete empirical interpolation method

The most natural approach to have an efficient implementation of the methodology would be to perform hyper-reduction directly on the residual term $\mathcal{R}$. In what follows we present a possible implementation using the discrete empirical interpolation method as originally introduced in [47]. During the offline stage there is the need also to store snapshots of the residual vector $\mathcal{R}$ and to assemble a snapshots matrix for the residual vector:

$$S_{\mathcal{R}} = [\mathcal{R}(t_1, \boldsymbol{\mu}_1), \mathcal{R}(t_2, \boldsymbol{\mu}_1), \ldots, \mathcal{R}(t_{N_t}, \boldsymbol{\mu}_{N_{\mu_{\text{train}}}}))] \in \mathbb{R}^{N_h \times N_s}. \tag{4.1}$$

Then the procedure requires to find a suitable basis $\boldsymbol{\Phi}_{\mathcal{R}}$ to interpolate the residual vector and a set of indices $i_{N_{\mathcal{R}}}$ where the interpolation condition is imposed. The basis can be constructed employing POD:

$$\boldsymbol{\Phi}_{\mathcal{R}} = [\phi_{\mathcal{R}_1}, \ldots, \phi_{\mathcal{R}_{N_{\mathcal{R}}}}] = \text{POD}(S_{\mathcal{R}}), \tag{4.2}$$

where the dimension of the POD space $N_{\mathcal{R}}$ is selected fixing a prescribed number of POD elements or a prescribed error on the projection error. The residual vector can be then approximated as:

$$\mathcal{R}(t, \mu) \approx \sum_{i=1}^{N_{\mathcal{R}}} a_i^{\mathcal{R}}(t, \mu) \phi_{\mathcal{R}_i}. \tag{4.3}$$

During the online stage, when a new value of the input parameter μ need to be tested, for a generic time instant t, it is necessary to compute the coefficient vector $a_i^{\mathcal{R}}$. This computation requires the evaluation of the residual $\mathcal{R}(t, \boldsymbol{\mu})$ only in the location identified by the indices $i_{N_{\mathcal{R}}}$. The expression to compute the coefficient vector $\boldsymbol{a}^{\mathcal{R}}$ is in fact given by:

$$\boldsymbol{a}^{\mathcal{R}}(t,\mu) = (\boldsymbol{P}^T \boldsymbol{\Phi}_{\mathcal{R}})\mathcal{R}_I(t,\mu) \quad \text{with} \quad \mathcal{R}_I(t,\mu) = \boldsymbol{P}^T \mathcal{R}(t,\mu), \tag{4.4}$$

where $\boldsymbol{P}$ and $\boldsymbol{\Phi}_{\mathcal{R}}$ are obtained following the procedure reported in algorithm 1. Therefore, the residual function needs to be evaluated only in a relatively small number of points (magic points) which is usually much smaller with respect to the total number of degrees of freedom used to discretize the domain.

Moreover, during the online stage, for a new untested geometry we need to check which magic points are marked as active and which ones are marked as inactive. However, this procedure scales with the number of magic points so it will not lead to a drastic increase of the online computational cost.

Remark 4.1. With respect to the implementation in a standard scenario with conforming meshes there is an additional complexity that we believe would require further studies. In the discrete empirical interpolation method, the residual is in fact interpolated using a basis obtained with snapshots having different sets of active and inactive nodes. It is therefore important to understand how to deal with the residual values on inactive nodes. We believe that the first step would be to study if a preprocessing of the residual snapshots matrix, similar to the one used for the snapshots, is beneficial or a "do nothing" approach is preferred.

Algorithm 1 The DEIM procedure

```
Input: snapshots matrix $S_{\mathcal{R}} = [\mathcal{R}(t_1, \boldsymbol{\mu}_1), \mathcal{R}(t_2, \boldsymbol{\mu}_1), \dots, \mathcal{R}(t_{N_t}, \boldsymbol{\mu}_{N_{\mu_{\text{train}}}}))]$, tolerance tol.
Output: DEIM basis Functions $\boldsymbol{\Phi}_{\mathcal{R}} = [\phi_{\mathcal{R}_1}, \dots, \phi_{\mathcal{R}_{N_{\mathcal{R}}}}]$, Interpolation indices $\boldsymbol{i}_{\Phi_{\mathcal{R}}} = [i_1, \dots, i_{N_{\mathcal{R}}}]$.
  compute the DEIM modes $\boldsymbol{\Phi}_{\mathcal{R}} = [\phi_{\mathcal{R}_1}, \dots, \phi_{\mathcal{R}_{N_{\mathcal{R}}}}] = \text{POD}(S_{\mathcal{R}})$
  $\varepsilon = \text{tol} + 1, \quad k = 1$
  $i_1 = \arg\max_{j=1,N_h} |(\phi_{\mathcal{R}_1})_j|$
  $\boldsymbol{\Phi}_{\mathcal{R}} = [\phi_{\mathcal{R}_1}], \quad \boldsymbol{i}_{N_{\mathcal{R}}} = [i_1], \quad \boldsymbol{P} = [\boldsymbol{e}_{i_1}]$
  while $\varepsilon > \text{tol}$ do
    $k = k + 1$
    Solve $(\boldsymbol{P}^T \boldsymbol{\Phi}_{\mathcal{R}})\boldsymbol{c} = \boldsymbol{P}^T \phi_{\mathcal{R}_k}$
    $\boldsymbol{r} = \phi_{\mathcal{R}_k} - \boldsymbol{\Phi}_{\mathcal{R}}\boldsymbol{c}$
    $i_k = \arg\max_{j=1,N_h} |(\boldsymbol{r})_j|$
    $\boldsymbol{\Phi}_{\mathcal{R}} = [\boldsymbol{\Phi}_{\mathcal{R}}, \phi_{\mathcal{R}_k}], \quad \boldsymbol{P} = [\boldsymbol{P}, \boldsymbol{e}_{i_k}], \quad \boldsymbol{i}_{N_{\mathcal{R}}} = [\boldsymbol{i}_{N_{\mathcal{R}}}, i_k]$
  end while
```

5. Numerical experiments

We consider three test cases featuring symmetric SWE flow past a stationary cylinder, based on the configuration in Figure 5.1. The straight channel is represented by the computational domain $\Omega = [-1.5,\ 1.5] \times [0.3,\ 0.3]$. Denoting the unit outer normal vector to $\partial\Omega$ by $\boldsymbol{n}$, the upper and lower boundary conditions are given by slippery walls, for which $\boldsymbol{v} \cdot \boldsymbol{n} = 0$, the left boundary is set to a constant inflow flux $h\boldsymbol{v} \cdot \boldsymbol{n} = -0.02$, and the right boundary condition is a constant outflow with flux $h\boldsymbol{v} \cdot \boldsymbol{n} = 0.02$. The flow direction is indicated by the arrows in the same figure. The cylinder location determined by two parameters, namely the radius R and the x-coordinate of its center x_c. In all computations, the initial condition is given by the uniform flow condition with $h = 0.2$ and $\boldsymbol{v} = (0.1,\ 0.0)$ at all active nodes. The first two test cases consist of geometrical parameterization using a one-dimensional parameter space with either x_c or R fixed, respectively. The third test case consists of a geometrical parameterization with a two-dimensional parameter space where both x_c and R are left free. All results for the test problems are obtained using a background triangular mesh with 5,419 vertices and 10,476 elements; the average element edge size is 0.02.

In all test cases, we generate the snapshots using FOM solutions that are obtained by a fixed CFL number $\alpha_{\text{cfl}} = 0.5$. Denoting the numerical solution at time t^n by $\boldsymbol{U}^n$ as before, then the snapshots are picked as $\boldsymbol{U}^{kn_{\text{freq}}}$,

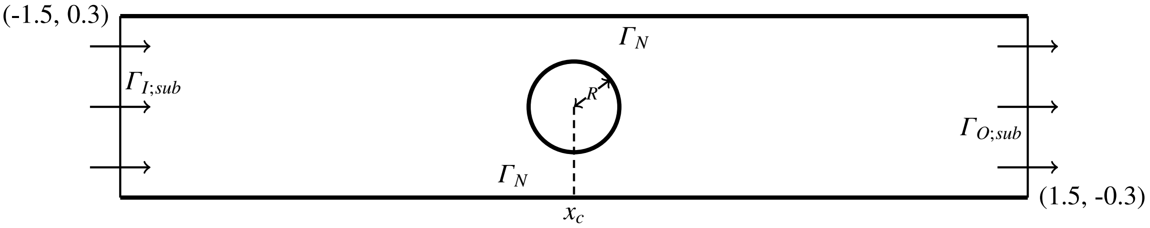


Figure 5.1: Symmetric SWE flow past a stationary cylinder in a straight channel, together with the two parameters characterizing the configuration in the numerical examples, namely the cylinder radius R and the x-coordinate of the cylinder center x_c.

$k = 1, 2, \cdots$, where n_{freq} is the sampling frequency; the solution at the terminal time is always picked as a snapshot, if not sampled already. To assess the performance of the reduced-order models, we instead compute both the ROM and FOM solutions using a fixed time step size $\Delta t = 0.002$, which corresponds to a CFL number slightly smaller than 0.5, to avoid interpolation error at different time steps.

For each test, we consider two sets of ROM computations: the first set uses the unprocessed POD basis vectors[1] and the second set uses POD basis vectors with interpolated values at inactive nodes. In all tables and plots, the two sets are designated by the standard SBM-ROM (without interpolation) and with interpolation (SBM-iROM), respectively. Given a prescribed energy threshold $\mathcal{E}_{\text{pod}}$, the number of POD modes used in the ROM computations is determined by the smallest number n such that:

$$\frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{N_s} \lambda_i} \geq \mathcal{E}_{\text{pod}} \ ,$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_s}$ are all the non-zero eigenvalues of the correlation matrix given in (3.2). As the test is convection dominated, we choose thresholds that are very close to unity, with the particular number of POD modes summarized in Table 2. In the first row of the table, we also indicate the sampling frequency n_{freq} for each test.

Table 2: Number of POD modes for each ROM computation in all three test cases. "SBM-iROM" and "SBM-ROM" stand for ROM with and without interpolation, respectively. The row with $\mathcal{E}_{\text{pod}} = 1$ corresponds to the sizes of full sets of POD modes.

	Test 1 ($n_{\text{freq}} = 2$)		Test 2 ($n_{\text{freq}} = 10$)		Test 3 ($n_{\text{freq}} = 10$)	
$\mathcal{E}_{\text{pod}}$	SBM-iROM	SBM-ROM	SBM-iROM	SBM-ROM	SBM-iROM	SBM-ROM
$1 - 10^{-5}$	15	16	63	67	83	107
$1 - 10^{-6}$	29	30	120	127	160	200
$1 - 10^{-7}$	52	53	178	181	275	350
$1 - 10^{-8}$	71	72	223	223	465	535
$1 - 10^{-9}$	85	86	256	256	646	679
1	561	561	344	344	1032	1032

The performance of each set of ROM computations is demonstrated both qualitatively and quantitatively:

- Qualitatively, the final water height solution is plotted and compared among the FOM computation and two ROM computations, for the latter the number of POD modes is determined by $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.
- Quantitatively, we compute and tabulate the relative error in Frobenius norm of ROM computations using the POD modes determined using the thresholds $\mathcal{E}_{\text{pod}} = 1 - 10^{-5}$, $1 - 10^{-6}$, $1 - 10^{-7}$, $1 - 10^{-8}$, and $1 - 10^{-9}$, and compare them to the projected FOM solutions (let the interpolated basis functions be $\boldsymbol{\Phi}_r$ as before, the projected

[1] That is, a constant value zero is filled at all inactive nodes in the snapshots, see Section 3.1 for more details.

solution of the full-order solution $\boldsymbol{U}_{\text{fom}}$ is given by $(\boldsymbol{\Phi}_r^T\boldsymbol{\Phi}_r)^{-1}\boldsymbol{\Phi}_r^T\boldsymbol{U}_{\text{fom}})$. In particular, the Frobenius norm of a generic solution vector $\boldsymbol{U} \in \mathbb{R}^{n_{\text{node}}}$ is defined as:

$$\|\boldsymbol{U}\|_{\text{Frob}(\tilde{\mathcal{D}})} \overset{\text{def}}{=\!=} \left(\sum_{A=1,\, \boldsymbol{x}_A \in \tilde{\mathcal{D}}}^{n_{\text{node}}} U_A^2 \right)^{\frac{1}{2}} . \tag{5.1}$$

To this end, the relative space-time Frobenius error of the ROM solution is computed as:

$$\left(\int_0^T \|\boldsymbol{U}_{\text{fom}} - \boldsymbol{U}_{\text{rom}}\|^2_{\text{Frob}(\tilde{\mathcal{D}})}\, dt \right)^{\frac{1}{2}} \Bigg/ \left(\int_0^T \|\boldsymbol{U}_{\text{fom}}\|^2_{\text{Frob}(\tilde{\mathcal{D}})}\, dt \right)^{\frac{1}{2}} ; \tag{5.2}$$

whereas the relative space-time Frobenius error of the projected FOM solution is given by:

$$\left(\int_0^T \left\|\boldsymbol{U}_{\text{fom}} - (\boldsymbol{\Phi}_r^T\boldsymbol{\Phi}_r)^{-1}\boldsymbol{\Phi}_r^T\boldsymbol{U}_{\text{fom}}\right\|^2_{\text{Frob}(\tilde{\mathcal{D}})}\, dt \right)^{\frac{1}{2}} \Bigg/ \left(\int_0^T \|\boldsymbol{U}_{\text{fom}}\|^2_{\text{Frob}(\tilde{\mathcal{D}})}\, dt \right)^{\frac{1}{2}} . \tag{5.3}$$

In both (5.2) and (5.3), the time integral is approximated by weighted sum of discrete solutions in a straight forward manner.

5.1. Test 1: Geometrical parameterization with varying cylinder radius

In this test case we fix $x_c = 0.0$ and generate the FOM snapshots using three radii $R = 0.1$, $R = 0.15$, and $R = 0.2$; in all tests the computation is performed until $T = 0.8$. A total number of 561 snapshots are created from the three FOM computations with a sampling frequency $n_{\text{freq}} = 2$, and we assess the performance of ROM by computing the flow past cylinders with radii $R = 0.08$, $R = 0.13$, $R = 0.17$, and $R = 0.22$.

Among the four, $R = 0.08$ is the most challenging one in the sense that several active nodes are inactive in all snapshots, hence it is not surprising to see that all ROM computations without interpolation fail to deliver any reasonable solutions. For the other three radii tested, all active nodes are also active in some snapshots and thus a solution is obtained whether the interpolation is employed or not; in these cases, we still observe that no interpolation leads to significantly worse ROM solution, as demonstrated by the height solution surfaces for the case $R = 0.13$ in Figure 5.2. In particular, SBM-ROM leads to an overall "shift" in the entire computational domain whereas the SBM-iROM solution agrees much better with the FOM one. It is worth noting here that in other two tests (that is when x_c is varying), the SBM-ROM solutions demonstrate oscillation with very large magnitude; hence one expects no significant improvement for ROM without interpolation, even if some non-zero value is used to fill all the inactive nodes in the snapshots.

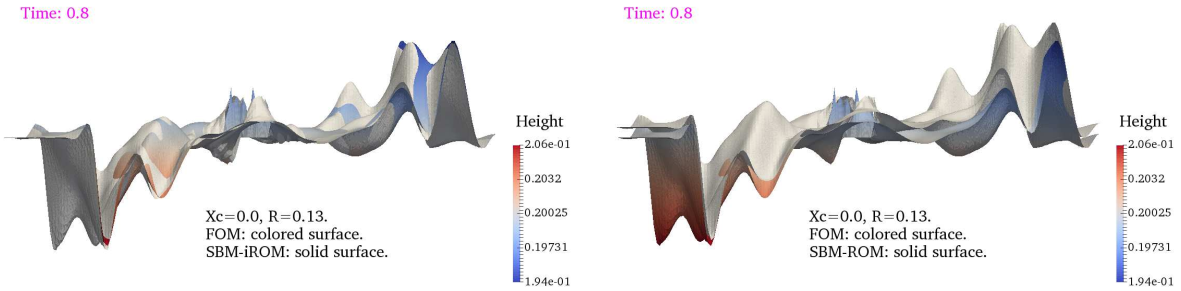


Figure 5.2: The solution surfaces for water height at $T = 0.8$ in the case $R = 0.13$ of Test 1 for SBM-iROM (left panel) and SBM-ROM (right panel). In both plots, the ROM solution (solid surface) is plotted on top of the FOM one (color surface).

Next in Figures 5.3–5.6, we plot the water height solutions at terminal time $T = 0.8$ in all four cases. As mentioned at the beginning of this section, the ROM computations are performed using POD modes corresponding to the energy threshold $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$, that is, 29 modes for SBM-iROM and 30 modes for SBM-ROM, see also Table 2. Note that

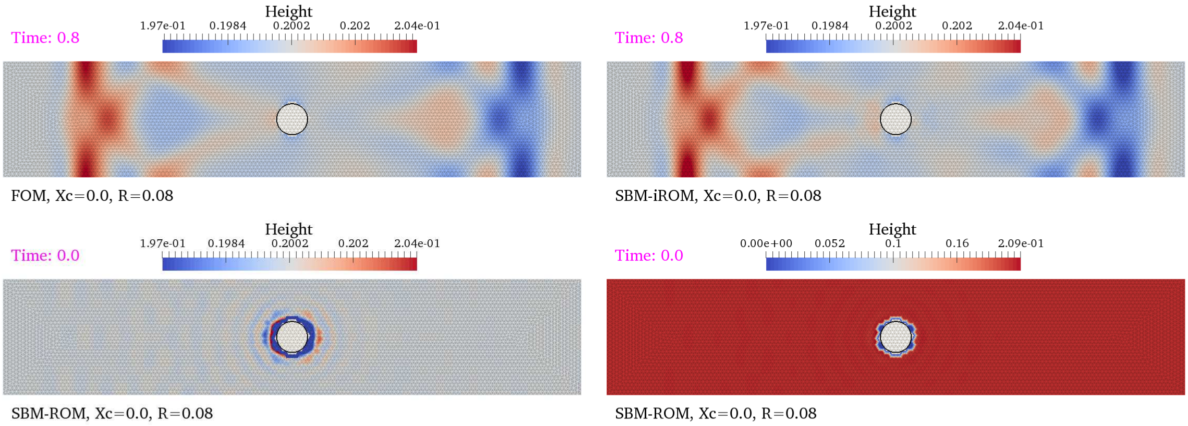


Figure 5.3: The height (h) in the case of $R = 0.08$ computed by FOM (top left), SBM-iROM (top right), and initial height of unsuccessful ROM computations without interpolation (bottom row). The legend range is set according to the FOM computation except the bottom right panel, where the SBM-ROM solution is plotted using its own range. All ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

when $R = 0.08$, SBM-ROM is unsuccessful and thus only the initial data is plotted (see the bottom row of Figure 5.3); furthermore, because the SBM-ROM solution is very different from the FOM one, we plot them both in the scale of the FOM solution and in the scale of its own. In comparison, when all active nodes in the ROM computation is captured by some snapshots, the SBM-ROM computations are successful and they produce fairly reasonable solutions, as shown in the bottom rows of Figures 5.4–5.6, where the SBM-ROM heights at the terminal time are plotted in both the FOM scale and its own scale, respectively.

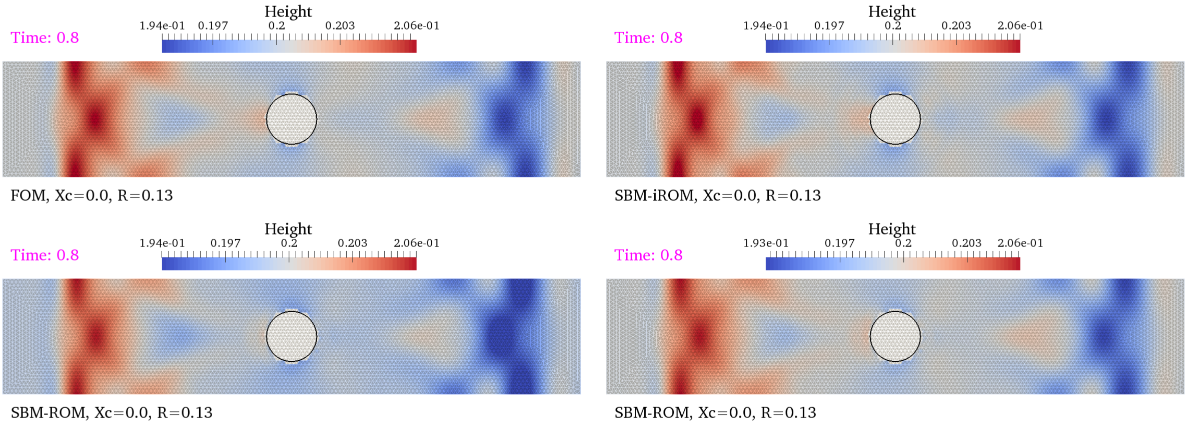


Figure 5.4: The height (h) in the case of $R = 0.13$ computed by FOM (top left), SBM-iROM (top right), and SBM-ROM (bottom row). The legend range is set according to the FOM computation except the bottom right panel, where the SBM-ROM solution is plotted using its own range. All ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

The relative space-time Frobenius errors of ROM computations using different number of POD modes and the projected FOM solutions are summarized in Table 3–Table 6 for the four radii, respectively. These errors are also plotted against the number of POD modes in Figure 5.7. The correspondence between the number of POD modes and $\mathcal{E}_{\text{pod}}$ is documented in Table 2. From these figures we see that the errors by SBM-iROM are very close to the projection error by the same set of POD modes, and these errors typically decrease as the number of modes increases. In contrast, without interpolation on the one hand the errors by the ROM solutions are far from the projection error, and they seem to increase as a larger number of POD modes is used.

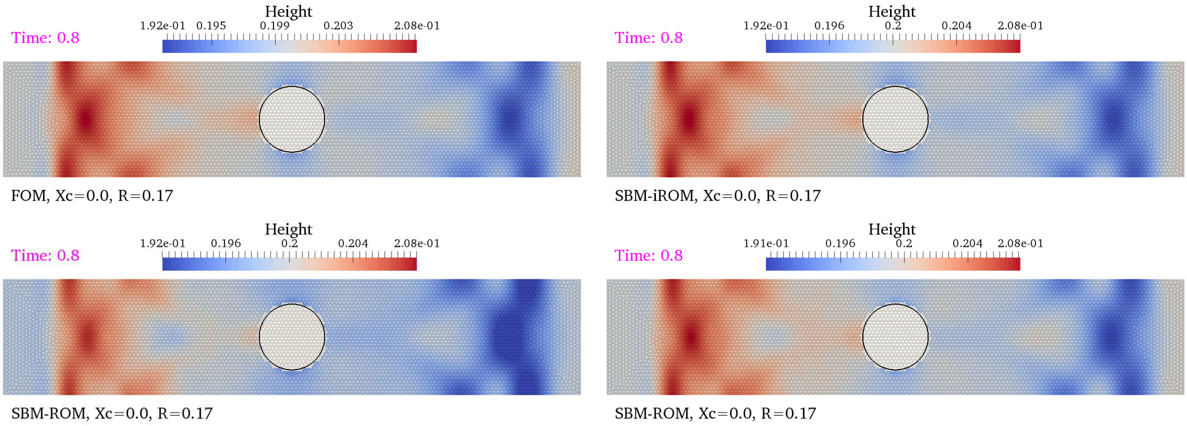


Figure 5.5: The height (h) in the case of $R = 0.17$ computed by FOM (top left), SBM-iROM (top right), and SBM-ROM (bottom row). The legend range is set according to the FOM computation except the bottom right panel, where the SBM-ROM solution is plotted using its own range. All ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

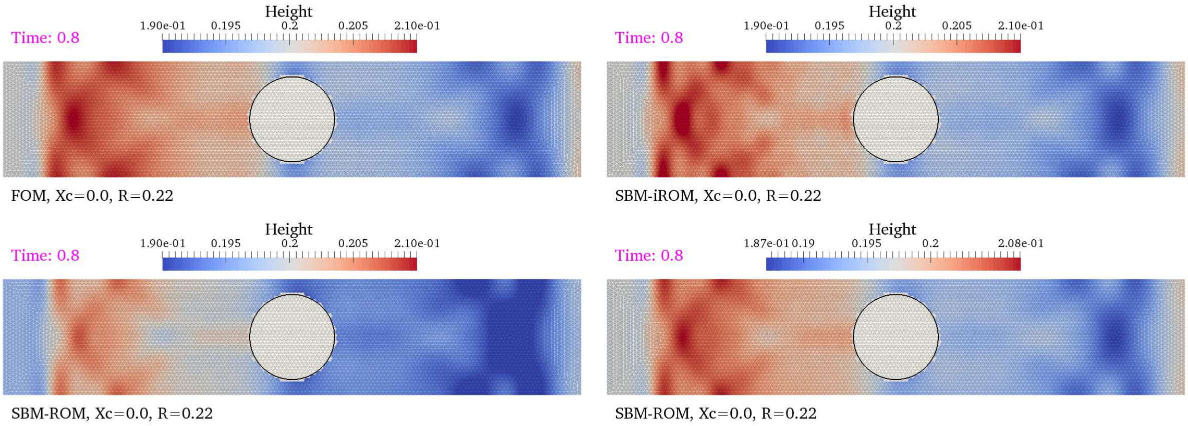


Figure 5.6: The height (h) in the case of $R = 0.22$ computed by FOM (top left), SBM-iROM (top right), and SBM-ROM (bottom row). The legend range is set according to the FOM computation except the bottom right panel, where the SBM-ROM solution is plotted using its own range. All ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

5.2. Test 2: Geometrical parameterization with varying cylinder center location

In the second test case we fix $R = 0.15$ and generate the FOM snapshots using three cylinder locations with $x_c = -0.5$, $x_c = 0.0$, and $x_c = 0.5$, with termination times being $T = 0.5$, $T = 0.8$, and $T = 0.5$, respectively. The shorter simulation period is selected to avoid interaction between the reflected waves and the left or right boundaries. A total number of 344 snapshots are created from the three FOM computations with a sampling frequency $n_{\text{freq}} = 10$, and we assess the performance of ROM by computing the flow past cylinders with the same radius and locations $x_c = -0.65$, $x_c = -0.15$, $x_c = 0.3$, and $x_c = 0.8$, with termination times given by $T = 0.4$, $T = 0.7$, $T = 0.6$, and $T = 0.3$, respectively.

With varying x_c, SBM-ROM is highly unstable, as demonstrated by the height solution surfaces for the case $x_c = -0.15$ in Figure 5.8; in contrast, the SBM-iROM solution agrees very well with the FOM one.

Next in Figures 5.9–5.12, we plot the water height solutions at terminal times using FOM and SBM-iROM; the SBM-ROM solutions are omitted as they are highly unstable, as also seen by the relative space-time Frobenius errors reported in Tables 7–10 and Figure 5.13.

Table 3: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.08$ of Test 1. Note that SBM-ROM fails to deliver any solution.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.53e-3	2.28e-2	2.12e-1	1.83e-3	3.21e-2	3.02e-1	–	–	–
$1 - 10^{-6}$	6.46e-4	1.44e-2	1.69e-1	1.30e-3	2.45e-2	2.14e-1	–	–	–
$1 - 10^{-7}$	3.88e-4	1.23e-2	1.54e-1	5.81e-4	1.84e-2	1.98e-1	–	–	–
$1 - 10^{-8}$	3.67e-4	1.19e-2	1.51e-1	5.39e-4	1.81e-2	1.98e-1	–	–	–
$1 - 10^{-9}$	3.59e-4	1.19e-2	1.50e-1	5.44e-4	1.81e-2	1.97e-1	–	–	–

Table 4: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.13$ of Test 1.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.23e-3	1.79e-2	9.81e-2	1.84e-3	2.69e-2	1.31e-1	5.82e-3	9.09e-2	3.97e-1
$1 - 10^{-6}$	4.75e-4	9.12e-3	6.40e-2	6.97e-4	1.32e-2	7.35e-2	3.51e-3	2.55e-2	1.13e-1
$1 - 10^{-7}$	2.56e-4	7.02e-3	5.97e-2	4.24e-4	1.14e-2	6.44e-2	3.87e-3	3.03e-2	1.46e-1
$1 - 10^{-8}$	2.38e-4	6.84e-3	5.85e-2	4.29e-4	1.14e-2	6.33e-2	4.67e-3	4.25e-2	2.12e-1
$1 - 10^{-9}$	2.29e-4	6.82e-3	5.80e-2	4.29e-4	1.13e-2	6.37e-2	4.96e-3	4.72e-2	2.31e-1

5.3. *Test 3: Geometrical parameterization study with two-dimensional parameter space*

In the last test case, we vary R the same way as in Test 1 and x_c the same way as in Test 2; hence there are 9 FOM computations with a total number of 1032 snapshots sampled with the frequency $n_{\text{freq}} = 10$. The ROM computations also follow the same variation in R and in x_c as in previous two test cases. In Figure 5.14 and Figure 5.15, we plot the height solution surfaces for the case $R = 0.08$, $x_c = -0.15$ and $R = 0.15$, $x_c = 0.3$, respectively. Note that as all nodes are active in some snapshots, SBM-ROM can handle $R = 0.08$ (c.f. Test 1), but the solution is highly unstable. The situation for SBM-ROM slightly improves when $R = 0.15$; nevertheless, in both demonstrations ROM with interpolation is significant superior to that without interpolation. Similar as in Test 2, we omit the solution plots for SBM-ROM, and compare the terminal water height solutions computed by FOM and SBM-iROM in Figures 5.16–5.19.

Lastly, we summarize the relative space-time Frobenius errors in Tables 11–22 and Figure 5.20.

6. Concluding remarks and future developments

In this article we have proposed and analyzed the coupling between the Shifted Boundary Method and POD-Galerkin methods for reduced order modelling in presence of geometrical parameters considering a case of hyperbolic systems. The methodology has been applied to shallow water equations discretized using an explicit time integration scheme and tested on three numerical benchmarks of increasing complexity.

In order to tackle one of the issues arising with the coupling of immersed methods and projection-based reduced-order models we proposed a new preprocessing technique applied on the full-order snapshots prior to the POD basis computation. This interpolation techniques demonstrates to increase considerably the accuracy of the results with respect to the unpreprocessed approach without any increase of the online computational costs.

The proposed preprocessing approach produces accurate results for cases that would otherwise yield completely unreliable results. This fact was particularly evident when the reduced-order model required to activate grid nodes that remained inactive for all training snapshots.

From a general remark, we observe that the coupling of immersed methods and projection based reduced models suffer from the necessity to use a large number of training points in order to produce accurate results during the online computations.

Table 5: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.17$ of Test 1.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.25e-3	1.79e-2	7.58e-2	1.60e-3	2.45e-2	9.07e-2	5.30e-3	2.45e-2	9.81e-2
$1 - 10^{-6}$	5.60e-4	1.04e-2	5.65e-2	6.90e-4	1.73e-2	7.14e-2	5.75e-3	2.81e-2	1.20e-1
$1 - 10^{-7}$	3.99e-4	8.84e-3	5.19e-2	5.45e-4	1.45e-2	6.57e-2	8.31e-3	7.65e-2	2.82e-1
$1 - 10^{-8}$	3.51e-4	8.38e-3	4.87e-2	5.28e-4	1.55e-2	6.70e-2	9.85e-3	1.02e-1	3.60e-1
$1 - 10^{-9}$	3.42e-4	8.34e-3	4.85e-2	5.09e-4	1.57e-2	6.73e-2	1.01e-2	1.08e-1	3.63e-1

Table 6: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.22$ of Test 1.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.20e-3	2.95e-2	9.05e-2	2.58e-3	3.67e-2	1.09e-1	1.60e-2	4.56e-2	1.15e-1
$1 - 10^{-6}$	9.71e-4	1.42e-2	4.76e-2	1.53e-3	2.38e-2	7.34e-2	1.60e-2	4.84e-2	1.09e-1
$1 - 10^{-7}$	5.55e-4	9.37e-3	3.70e-2	1.02e-3	1.63e-2	4.63e-2	1.60e-2	7.46e-2	2.04e-1
$1 - 10^{-8}$	4.78e-4	8.74e-3	3.52e-2	7.40e-4	1.15e-2	4.28e-2	1.58e-2	7.62e-2	2.39e-1
$1 - 10^{-9}$	4.57e-4	8.55e-3	3.47e-2	8.02e-4	1.16e-2	4.41e-2	1.51e-2	7.39e-2	2.00e-1

In future works we aim to study possible approaches in order to circumvent this limitation applying suitable transformations to the POD basis. A promising approach could be the preprocessing of the solution snapshots employing suitable non-linear transformation using neural networks [49] or other types of transformations [50]. In order to speed-up the online computations, other interesting aspects are related to the hyper-reduction of the problem using residual evaluations only in a limited number of nodes. Despite the fact that hyper-reduction techniques are now well developed, their application to immersed methods might be challenging.

We also plan to extend the proposed methodologies to fluid structure interaction problems.

Acknowledgments

This research has been supported by the Army Research Office (ARO) under Grant W911NF-18-1-0308 (GS), the U.S. National Science Foundation under Grant DMS-2137934, European Union Funding for Research and Innovation -Horizon 2020 Program- in the framework of European Research Council Executive Agency: Consolidator Grant H2020 ERC CoG 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics” (PI Prof. Gianluigi Rozza). We also acknowledge the INDAM-GNCS project “Tecniche Numeriche Avanzate per Applicazioni Industriali”, and by project FSE-European Social Fund-HEaD “Higher Education and Development” SISSA operazione 1, Regione Autonoma Friuli-Venezia Giulia, the HFRI and GSRT under grant agreement No 1115, the “First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment” grant 3270, and the support of the National Infrastructures for Research and Technology S.A. (GRNET S.A.) under project ID pa190902.

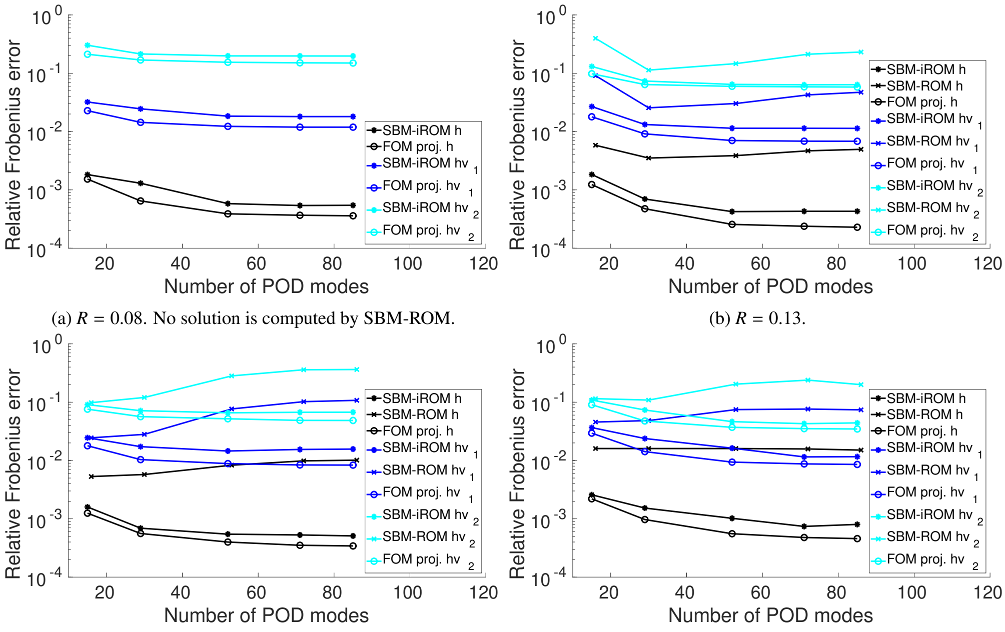
References

[1] P. Benner, M. Ohlberger, A. Patera, G. Rozza, K. Urban, Model Reduction of Parametrized Systems, Vol. 17 of MS&A series, Springer, 2017.

[2] Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms, Berlin, Boston, De Gruyter, 2020, https://doi.org/10.1515/9783110671490.

[3] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.

[4] A. Main, G. Scovazzi, The Shifted Boundary Method for embedded domain computations. Part I: Poisson and Stokes problems, Journal of Computational Physics 372 (2018) 972–995.



(a) $R = 0.08$. No solution is computed by SBM-ROM.

(b) $R = 0.13$.

(c) $R = 0.17$.

(d) $R = 0.22$.

Figure 5.7: The relative Frobenius errors of Test 1 by SBM-iROM (marked by $*$), SBM-ROM (marked by x), and FOM projection error (FOM proj., marked by o) in logarithmic scale plotted against the number of POD modes. The errors in h, hv_1, hv_2 are plotted in black, blue, and cyan colors, respectively.

Table 7: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $x_c = -0.65$ of Test 2.

	FOM projection			SBM-iROM			SBM-ROM		
ε_{pod}	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.33e-3	2.78e-2	1.95e-1	3.17e-3	4.53e-2	2.72e-1	6.56e-2	4.32e-1	1.82e+0
$1 - 10^{-6}$	1.40e-3	1.84e-2	1.48e-1	2.05e-3	3.13e-2	1.87e-1	6.10e-2	4.39e-1	1.90e+0
$1 - 10^{-7}$	1.21e-3	1.60e-2	1.26e-1	1.90e-3	2.99e-2	1.88e-1	6.11e-2	4.03e-1	1.89e+0
$1 - 10^{-8}$	1.17e-3	1.46e-2	1.18e-1	1.86e-3	2.93e-2	1.77e-1	5.94e-2	3.93e-1	1.87e+0
$1 - 10^{-9}$	1.16e-3	1.45e-2	1.15e-1	1.77e-3	2.86e-2	1.80e-1	5.72e-2	3.74e-1	1.81e+0

[5] A. Main, G. Scovazzi, The Shifted Boundary Method for embedded domain computations. Part II: Linear advection-diffusion and incompressible Navier–Stokes equations, Journal of Computational Physics 372 (2018) 996–1026.

[6] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The Shifted Boundary Method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, J. Comput. Phys. 369 (2018) 45–79.

[7] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, J. Appl. Mech. 70 (1) (2003) 10–17.

[8] A. Gerstenberger, W. A. Wall, An eXtended finite element method/lagrange multiplier based approach for fluid-structure interaction, Comput. Methods Appl. Mech. Eng. 197 (19–20) (2008) 1699–1714.

[9] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, Int. J. Numer. Meth. Eng. 84 (3) (2010) 253–304.

[10] G. Stabile, M. Zancanaro, G. Rozza, Efficient Geometrical parametrization for finite-volume based reduced order methods, International Journal for Numerical Methods in Engineering 121 (12) (2020) 2655–2682.

[11] M. Tezzele, N. Demo, G. Stabile, A. Mola, G. Rozza, Enhancing CFD predictions in shape design problems by model and parameter space reduction, Advanced Modeling and Simulation in Engineering Sciences 7 (40), (2020).

[12] N. M. Atallah, C. Canuto, G. Scovazzi, Analysis of the Shifted Boundary Method for the Poisson problem in domains with corners, Math. Comp. 90, (2021).

[13] E. N. Karatzas, G. Stabile, N. Atallah, G. Scovazzi, G. Rozza, A Reduced Order Approach for the Embedded Shifted boundary FEM and a Heat Exchange System on Parametrized Geometries, In: Fehr J., Haasdonk B. (eds) IUTAM Symposium on Model Order Reduction of Coupled Systems, Stuttgart, Germany, May 22–25, 2018. IUTAM Bookseries, vol 36. Springer, Cham (2020).

[14] E. N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A reduced basis approach for PDEs on parametrized geometries based on

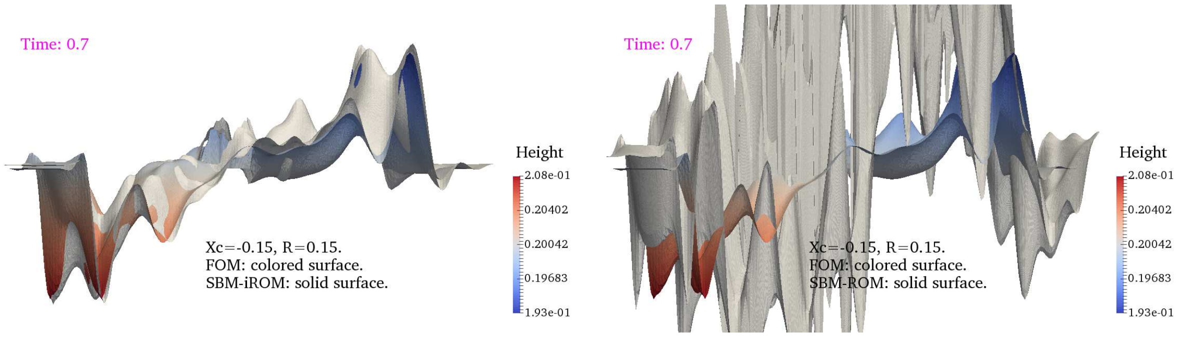


Figure 5.8: The solution surfaces for water height at $T = 0.7$ in the case $x_c = -0.15$ of Test 2 for SBM-iROM (left panel) and SBM-ROM (right panel). In both plots, the ROM solution (solid surface) is plotted on top of the FOM one (color surface).

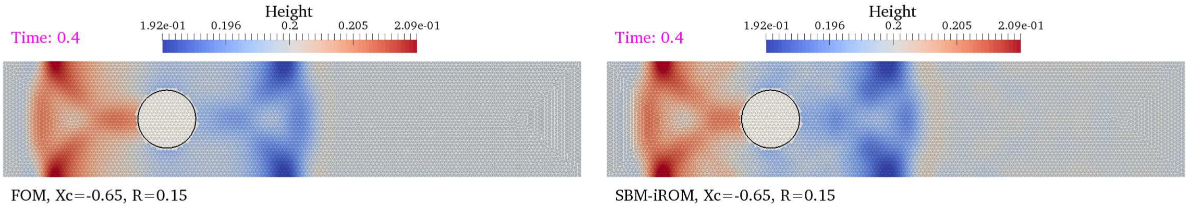


Figure 5.9: The height (h) in the case of $x_c = -0.65$ computed by FOM (left panel) and SBM-iROM (right panel), with the legend range set according to the FOM computation. The ROM computation is performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

the shifted boundary finite element method and application to a Stokes flow, Computer Methods in Applied Mechanics and Engineering 347 (2019) 568–587.

[15] E. N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi, G. Rozza, A reduced-order shifted boundary method for parametrized incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 370 (2020) 113–273.

[16] E. N. Karatzas, G. Rozza, A Reduced Order Model for a Stable Embedded Boundary Parametrized Cahn-Hilliard Phase-Field System Based on Cut Finite Elements, J Sci Comput 89 (9), (2021), https://doi.org/10.1007/s10915-021-01623-8.

[17] G. Katsouleas, E. N. Karatzas, F. Travlopanos, Discrete empirical interpolation and unfitted mesh FEMs: application in PDE-constrained optimization, Optimization (2022) 1–34, https://doi.org/10.1080/02331934.2022.2032697.

[18] A. Nouy, M. Chevreuil, E. Safatly, Fictitious domain method and separated representations for the solution of boundary value problems on uncertain parameterized domains, Computer Methods in Applied Mechanics and Engineering 200 (45-46) (2011) 3066–3082.

[19] M. Balajewicz, C. Farhat, Reduction of nonlinear embedded boundary models for problems with evolving interfaces, Journal of Computational Physics 274 (2014) 489–504.

[20] Y. Kagawa, T. Tsuchiya, T. Yamabuchi, H. Kawabe, T. Fujii, Finite element simulation of non-linear sound wave propagation, Journal of Sound and Vibration 154 (1) (1992) 125–145.

[21] G. Scovazzi, B. Carnes, Weak boundary conditions for wave propagation problems in confined domains: Formulation and implementation using a variational multiscale method, Computer Methods in Applied Mechanics and Engineering 221-222 (2012) 117–131.

[22] 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, Computer Methods in Applied Mechanics and Engineering 325 (2017) 532–576.

[23] T. Song, G. Scovazzi, A Nitsche method for wave propagation problems in time domain, Comput. Methods Appl. Mech. Eng. 293 (2015) 481–521.

[24] G. Hauke, A symmetric formulation for computing transient shallow water flows, Computer Methods in Applied Mechanics and Engineering 163 (1) (1998) 111–122.

[25] G. Hauke, A stabilized finite element method for the saint-venant equations with application to irrigation, International Journal for Numerical Methods in Fluids 38 (10) (2002) 963–984.

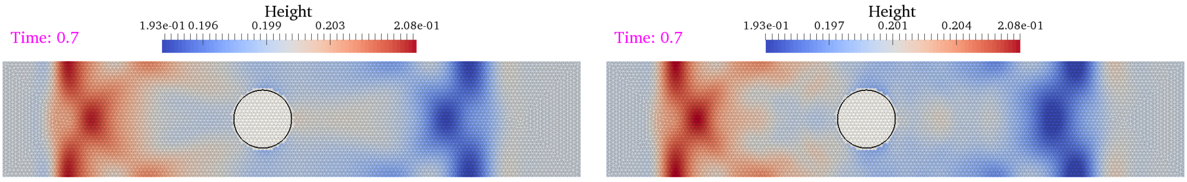
[26] K. Kashiyama, H. Ito, M. Behr, T. Tezduyar, Three-step explicit finite element computation of shallow water flows on a massively parallel computer, International Journal for Numerical Methods in Fluids 21 (10) (1995) 885–900.

[27] M. Kawahara, K. Kashiyama, Selective lumping finite element method for nearshore current, International Journal for Numerical Methods in Fluids 4 (1) (1984) 71–97.

[28] M. Kawahara, K. Kashiyama, Boundary type finite element method for surface wave motion based on trigonometric function interpolation, International Journal for Numerical Methods in Engineering 21 (10) (1985) 1833–1852.

[29] K. Kashiyama, M. Kawahara, A boundary-type finite element model for water surface wave problems, International Journal for Numerical Methods in Fluids 8 (1) (1988) 65–79.

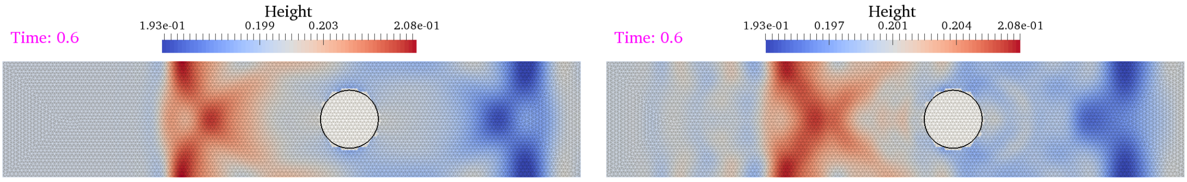
[30] K. Kashiyama, K. Saitoh, M. Behr, T. Tezduyar, Parallel finite element methods for large-scale computation of storm surges and tidal flows, International Journal for Numerical Methods in Fluids 24 (12) (1997) 1371–1389.



FOM, Xc=-0.15, R=0.15

SBM-iROM, Xc=-0.15, R=0.15

Figure 5.10: The height (h) in the case of $x_c = -0.15$ computed by FOM (left panel) and SBM-iROM (right panel), with the legend range set according to the FOM computation. The ROM computation is performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.



FOM, Xc=0.3, R=0.15

SBM-iROM, Xc=0.3, R=0.15

Figure 5.11: The height (h) in the case of $x_c = 0.3$ computed by FOM (left panel) and SBM-iROM (right panel), with the legend range set according to the FOM computation. The ROM computation is performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

[31] K. Kazuo, S. Masaaki, Adaptive boundary-type finite element method for wave diffraction-refraction in harbors, Computer Methods in Applied Mechanics and Engineering 112 (1) (1994) 185–197.

[32] M. Kawahara, H. Sakurai, K. Kashiyama, Boundary-type finite element method for wave propagation analysis, International Journal for Numerical Methods in Fluids 8 (5) (1988) 559–578.

[33] M. Kawahara, N. Takeuchi, T. Yoshida, Two step explicit finite element method for tsunami wave propagation analysis, International Journal for Numerical Methods in Engineering 12 (2) (1978) 331–351.

[34] M. Ricchiuto, A. Bollermann, Stabilized residual distribution for shallow water simulations, Journal of Computational Physics 228 (4) (2009) 1071–1115.

[35] S. Takase, K. Kashiyama, S. Tanaka, T. E. Tezduyar, Space-time supg formulation of the shallow-water equations, International Journal for Numerical Methods in Fluids 64 (10-12) (2010) 1379–1394.

[36] W. Smith, The application of finite element analysis to body wave propagation problems, Geophys. J. R. Astr. Soc. 42 (2) (1975) 747–768.

[37] A. J. C. Barré de Saint-Venant, Théorie du mouvement non permanent des eaux, avec application aux crues des riviéres et à l'introduction des marées dans leur lit (1871) 11 p.Académie des sciences (France), Extrait des Comptes rendus de l'Académie des Sciences, tome LXXIII, séances des 17 et 24 juillet 1871.

[38] G. Kounadis, V. Dougalis, Galerkin finite element methods for the shallow water equations over variable bottom, Journal of Computational and Applied Mathematics 373 (2020) 112315.

[39] C. B. Vreugdenhil, Numerical Methods for Shallow-Water Flow, Water Science and Technology Library, 1994.

[40] E. Tadmor, Skew-selfadjoint form for systems of conservation laws, J. Math. Anal. Appl. 103 (2) (1984) 428–442.

[41] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The Shifted Boundary Method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, Inria Bordeaux Sud-Ouest RR-9136 (2017) 1–56.

[42] L. Sirovich, Turbulence and the dynamics of coherent structures, Parts I-III, Quarterly of applied mathematics (1987) 561–590.

[43] E. N. Karatzas, F. Ballarin, G. Rozza, Projection-based reduced order models for a cut finite element method in parametrized domains, Computers & Mathematics with Applications 79 (3) (2020) 833–851.

[44] M. D. Buhmann, Radial basis functions: theory and implementations, Cambridge University Press, Cambridge New York, 2003.

[45] G. E. Fasshauer, Meshfree Approximation Methods with Matlab, World Scientific, 2007.

[46] M. Barrault, Y. Maday, N. Nguyen, A. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique 339 (9) (2004) 667–672.

[47] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[48] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics 242 (2013) 623 – 647.

[49] D. Papapicco, N. Demo, M. Girfoglio, G. Stabile, G. Rozza, The neural network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations, Computer Methods in Applied Mechanics and Engineering 392 (2022) 114687.

[50] J. Ren, W. R. Wolf, X. Mao, Model reduction of traveling-wave problems via radon cumulative distribution transform, Physical Review Fluids 6 (8) (2021) L082501.

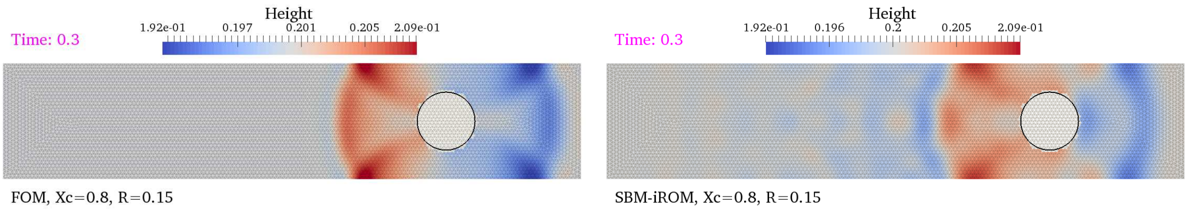


Figure 5.12: The height (h) in the case of $x_c = 0.8$ computed by FOM (left panel) and SBM-iROM (right panel), with the legend range set according to the FOM computation. The ROM computation is performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

Table 8: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $x_c = -0.15$ of Test 2.

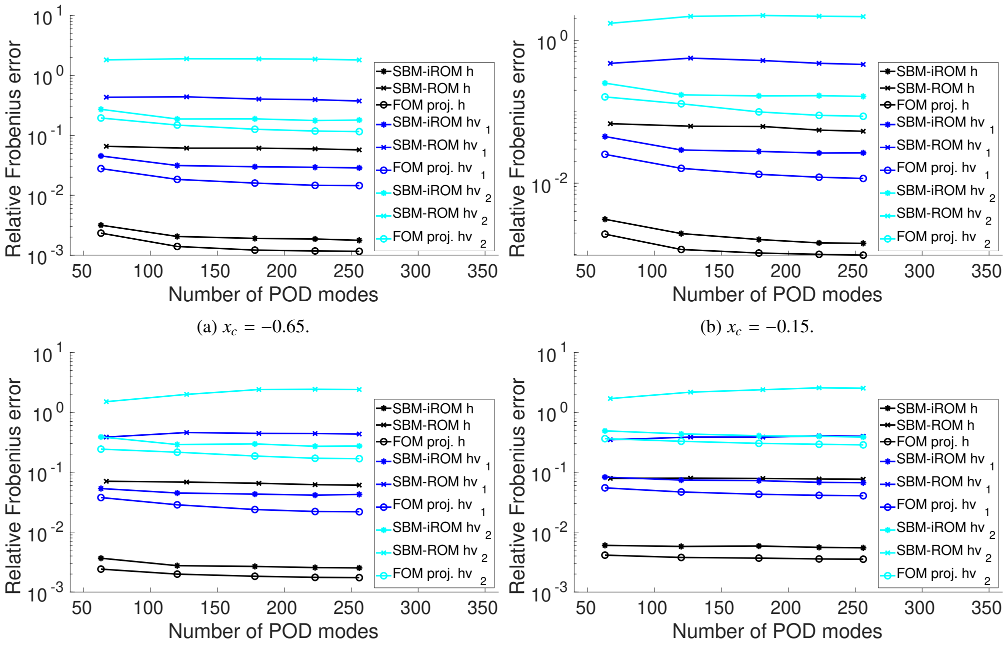
	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.93e-3	2.53e-2	1.61e-1	3.12e-3	4.50e-2	2.51e-1	6.80e-2	4.76e-1	1.73e+0
$1 - 10^{-6}$	1.18e-3	1.61e-2	1.29e-1	1.97e-3	2.91e-2	1.72e-1	6.27e-2	5.63e-1	2.17e+0
$1 - 10^{-7}$	1.05e-3	1.33e-2	9.94e-2	1.63e-3	2.78e-2	1.67e-1	6.21e-2	5.21e-1	2.23e+0
$1 - 10^{-8}$	1.01e-3	1.21e-2	8.90e-2	1.46e-3	2.64e-2	1.68e-1	5.53e-2	4.76e-1	2.17e+0
$1 - 10^{-9}$	9.82e-4	1.16e-2	8.64e-2	1.44e-3	2.66e-2	1.64e-1	5.33e-2	4.60e-1	2.15e+0

Table 9: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $x_c = 0.3$ of Test 2.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.42e-3	3.78e-2	2.43e-1	3.67e-3	5.34e-2	3.88e-1	7.06e-2	3.86e-1	1.50e+0
$1 - 10^{-6}$	1.99e-3	2.86e-2	2.16e-1	2.76e-3	4.50e-2	2.90e-1	6.85e-2	4.59e-1	1.99e+0
$1 - 10^{-7}$	1.84e-3	2.38e-2	1.86e-1	2.68e-3	4.31e-2	2.97e-1	6.54e-2	4.45e-1	2.40e+0
$1 - 10^{-8}$	1.76e-3	2.21e-2	1.71e-1	2.56e-3	4.15e-2	2.71e-1	6.20e-2	4.43e-1	2.42e+0
$1 - 10^{-9}$	1.74e-3	2.19e-2	1.68e-1	2.53e-3	4.26e-2	2.74e-1	6.09e-2	4.34e-1	2.40e+0

Table 10: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $x_c = 0.8$ of Test 2.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	4.13e-3	5.48e-2	3.63e-1	6.02e-3	8.29e-2	4.90e-1	7.90e-2	3.48e-1	1.69e+0
$1 - 10^{-6}$	3.78e-3	4.68e-2	3.28e-1	5.77e-3	7.39e-2	4.37e-1	7.96e-2	3.86e-1	2.17e+0
$1 - 10^{-7}$	3.69e-3	4.29e-2	3.03e-1	5.87e-3	7.24e-2	4.07e-1	7.86e-2	3.85e-1	2.38e+0
$1 - 10^{-8}$	3.57e-3	4.11e-2	2.93e-1	5.57e-3	6.78e-2	3.98e-1	7.72e-2	4.03e-1	2.56e+0
$1 - 10^{-9}$	3.54e-3	4.05e-2	2.86e-1	5.48e-3	6.69e-2	3.86e-1	7.64e-2	3.99e-1	2.52e+0



(a) $x_c = -0.65$.

(b) $x_c = -0.15$.

(c) $x_c = 0.3$.

(d) $x_c = 0.8$.

Figure 5.13: The relative Frobenius errors of Test 2 by SBM-iROM (marked by $*$), SBM-ROM (marked by x), and FOM projection error (FOM proj., marked by o) in logarithmic scale plotted against the number of POD modes. The errors in h, hv_1, hv_2 are plotted in black, blue, and cyan colors, respectively.

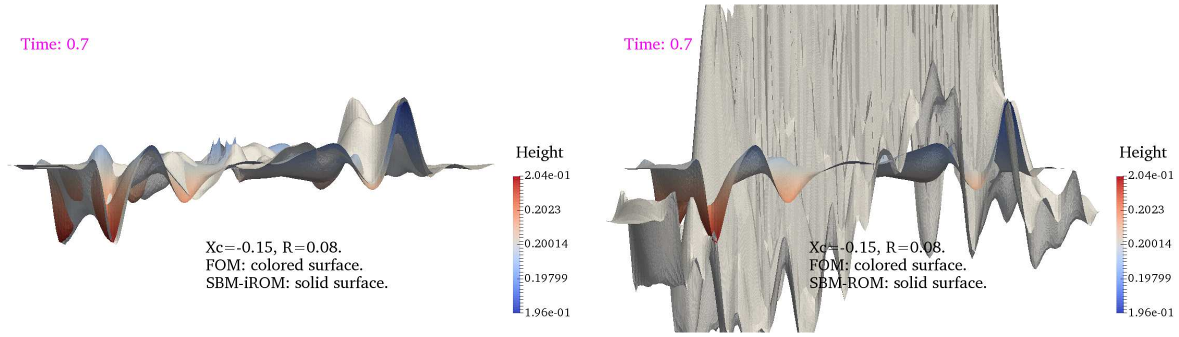


Figure 5.14: The solution surfaces for water height at $T = 0.7$ in the case $R = 0.08$ and $x_c = -0.15$ of Test 3 for SBM-iROM (left panel) and SBM-ROM (right panel). In both plots, the ROM solution (solid surface) is plotted on top of the FOM one (color surface).

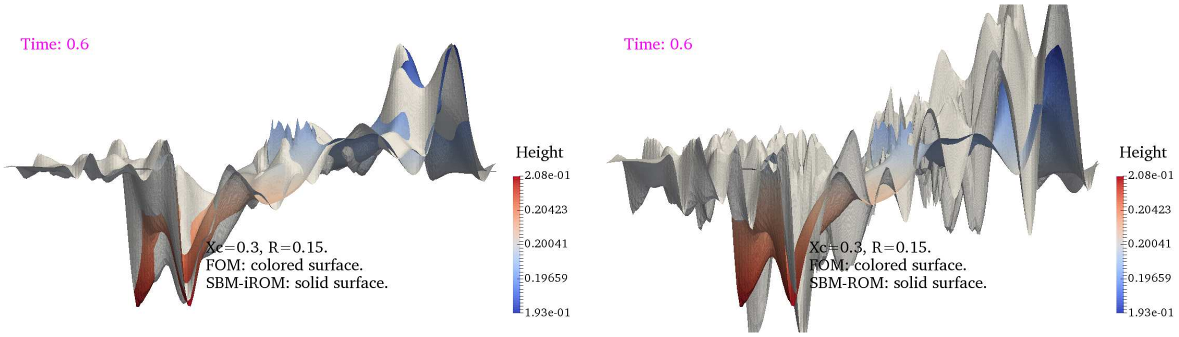
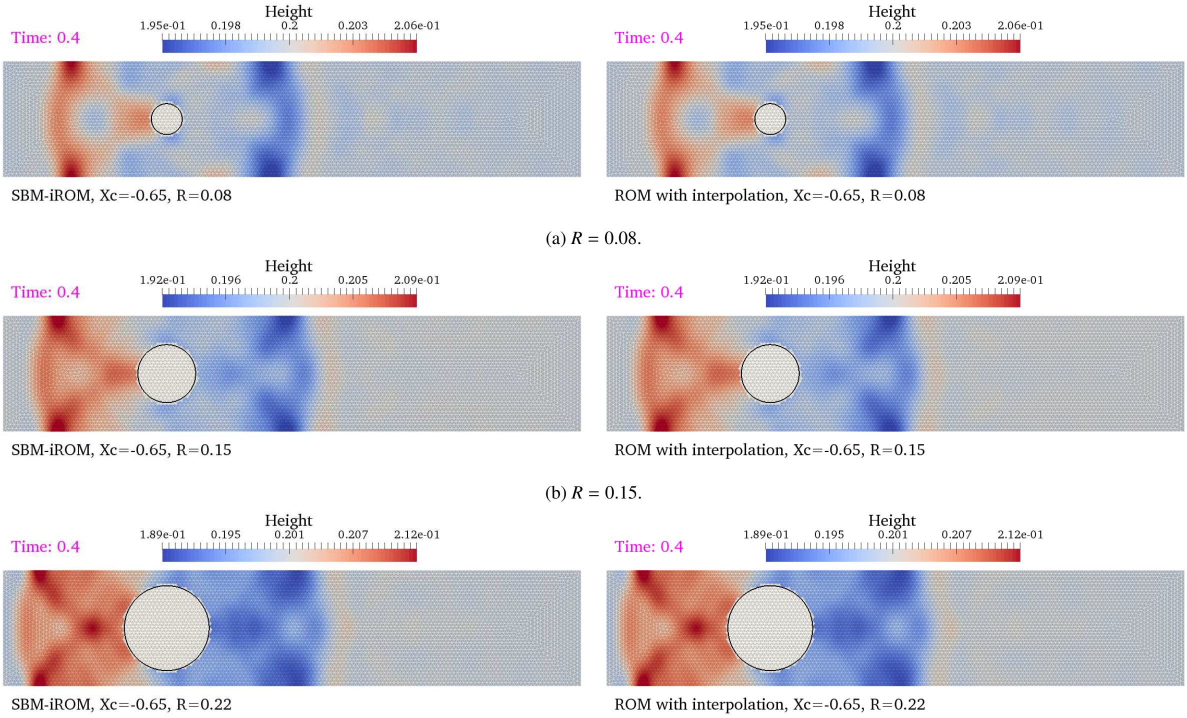


Figure 5.15: The solution surfaces for water height at $T = 0.6$ in the case $R = 0.15$ and $x_c = 0.3$ of Test 3 for SBM-iROM (left panel) and SBM-ROM (right panel). In both plots, the ROM solution (solid surface) is plotted on top of the FOM one (color surface).



(a) $R = 0.08$.

(b) $R = 0.15$.

(c) $R = 0.22$.

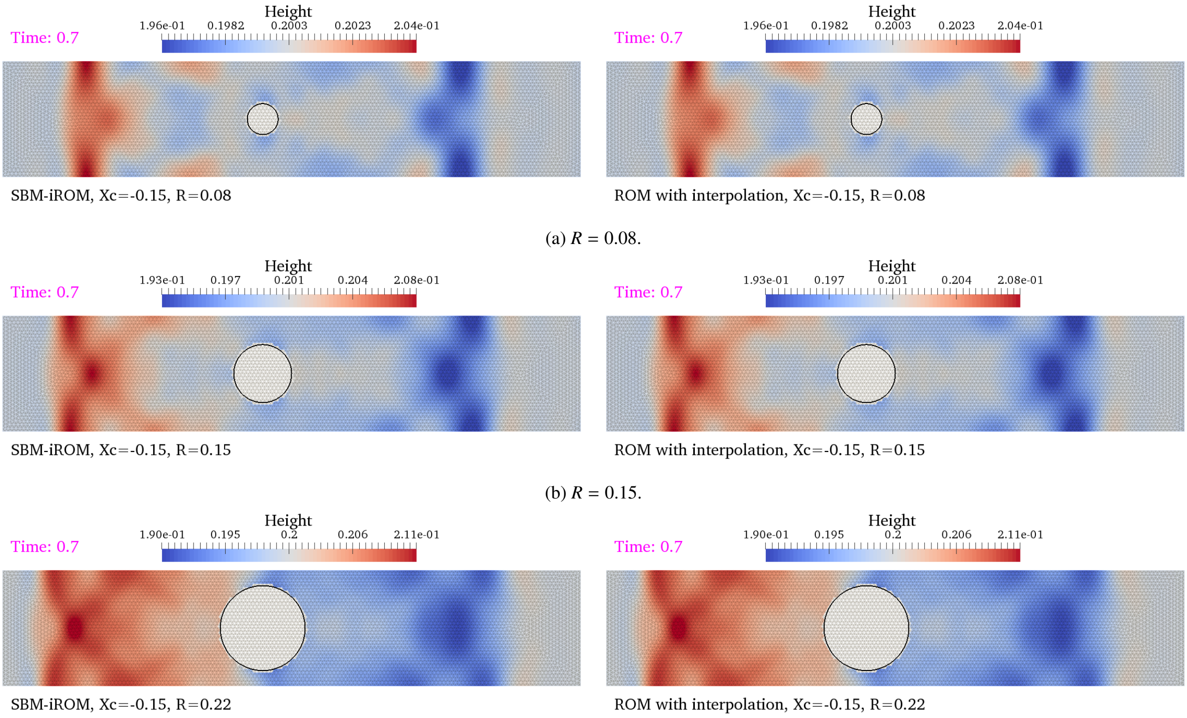
Figure 5.16: The height (h) in the case of $R = 0.08$ (top row), $R = 0.15$ (middle row), or $R = 0.22$ (bottom row), and $x_c = -0.65$ computed by FOM (left panels) and SBM-iROM (right panels), with the legend range set according to the FOM computations. The ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

Table 11: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.08$ and $x_c = -0.65$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.54e-3	3.26e-2	3.91e-1	2.85e-3	5.62e-2	5.84e-1	5.26e-2	2.70e-1	2.16e+0
$1 - 10^{-6}$	1.12e-3	2.61e-2	3.19e-1	2.14e-3	4.52e-2	4.85e-1	5.14e-2	2.75e-1	2.47e+0
$1 - 10^{-7}$	9.83e-4	1.92e-2	2.24e-1	1.67e-3	4.04e-2	4.20e-1	3.60e-2	3.61e-1	5.76e+0
$1 - 10^{-8}$	7.49e-4	1.39e-2	1.63e-1	9.65e-4	2.93e-2	3.10e-1	2.69e-2	2.66e-1	2.81e+0
$1 - 10^{-9}$	6.85e-4	1.15e-2	1.31e-1	8.53e-4	2.60e-2	2.58e-1	2.63e-2	2.79e-1	2.65e+0

Table 12: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.15$ and $x_c = -0.65$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.10e-3	2.75e-2	1.71e-1	3.37e-3	5.01e-2	2.74e-1	1.28e-2	1.56e-1	7.46e-1
$1 - 10^{-6}$	1.25e-3	1.83e-2	1.27e-1	1.82e-3	3.19e-2	1.90e-1	1.22e-2	1.38e-1	7.64e-1
$1 - 10^{-7}$	7.10e-4	8.25e-3	7.28e-2	1.10e-3	2.03e-2	1.33e-1	1.05e-2	1.22e-1	6.58e-1
$1 - 10^{-8}$	3.91e-4	5.10e-3	4.18e-2	6.49e-4	1.19e-2	8.47e-2	9.78e-3	1.11e-1	6.37e-1
$1 - 10^{-9}$	3.06e-4	4.14e-3	3.48e-2	5.05e-4	1.01e-2	6.99e-2	8.91e-3	1.03e-1	5.78e-1



(a) $R = 0.08$.

(b) $R = 0.15$.

(c) $R = 0.22$.

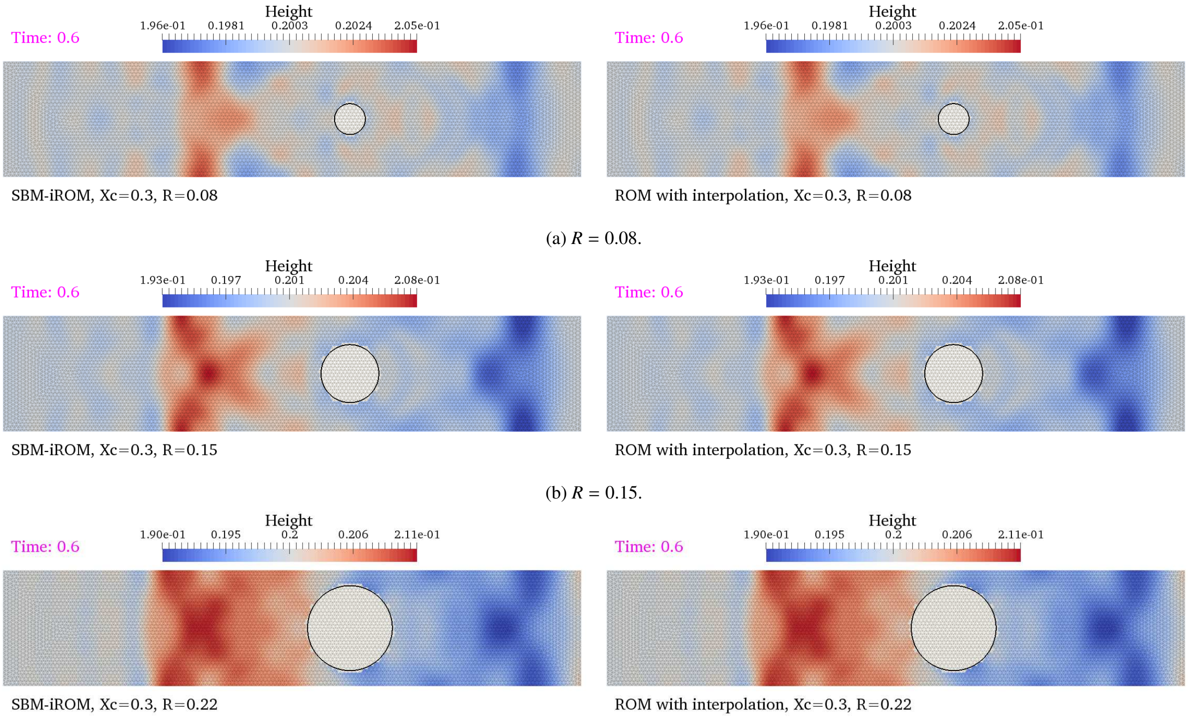
Figure 5.17: The height (h) in the case of $R = 0.08$ (top row), $R = 0.15$ (middle row), or $R = 0.22$ (bottom row), and $x_c = -0.15$ computed by FOM (left panels) and SBM-iROM (right panels), with the legend range set according to the FOM computations. The ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

Table 13: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.22$ and $x_c = -0.65$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.76e-3	2.93e-2	1.50e-1	3.90e-3	5.27e-2	2.69e-1	1.85e-2	2.09e-1	8.46e-1
$1 - 10^{-6}$	1.68e-3	1.93e-2	1.14e-1	2.45e-3	3.39e-2	2.08e-1	1.80e-2	2.11e-1	8.75e-1
$1 - 10^{-7}$	7.22e-4	7.85e-3	5.48e-2	1.10e-3	1.64e-2	9.50e-2	1.77e-2	2.06e-1	9.13e-1
$1 - 10^{-8}$	4.14e-4	4.83e-3	3.34e-2	7.16e-4	1.14e-2	6.62e-2	1.71e-2	1.99e-1	8.37e-1
$1 - 10^{-9}$	2.87e-4	3.84e-3	2.49e-2	5.67e-4	8.76e-3	4.91e-2	1.62e-2	1.97e-1	7.60e-1

Table 14: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.08$ and $x_c = -0.15$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.33e-3	2.96e-2	3.63e-1	2.47e-3	5.05e-2	6.32e-1	5.38e-2	3.06e-1	1.99e+0
$1 - 10^{-6}$	9.88e-4	2.37e-2	3.14e-1	2.04e-3	4.90e-2	4.72e-1	5.20e-2	3.62e-1	2.40e+0
$1 - 10^{-7}$	1.05e-3	1.81e-2	2.28e-1	1.36e-3	4.57e-2	4.35e-1	3.58e-2	4.57e-1	6.01e+0
$1 - 10^{-8}$	6.71e-4	1.32e-2	1.45e-1	8.21e-4	3.61e-2	3.12e-1	2.57e-2	3.14e-1	2.76e+0
$1 - 10^{-9}$	5.72e-4	1.07e-2	1.17e-1	6.48e-4	2.72e-2	2.46e-1	2.39e-2	3.32e-1	2.55e+0



(a) $R = 0.08$.

(b) $R = 0.15$.

(c) $R = 0.22$.

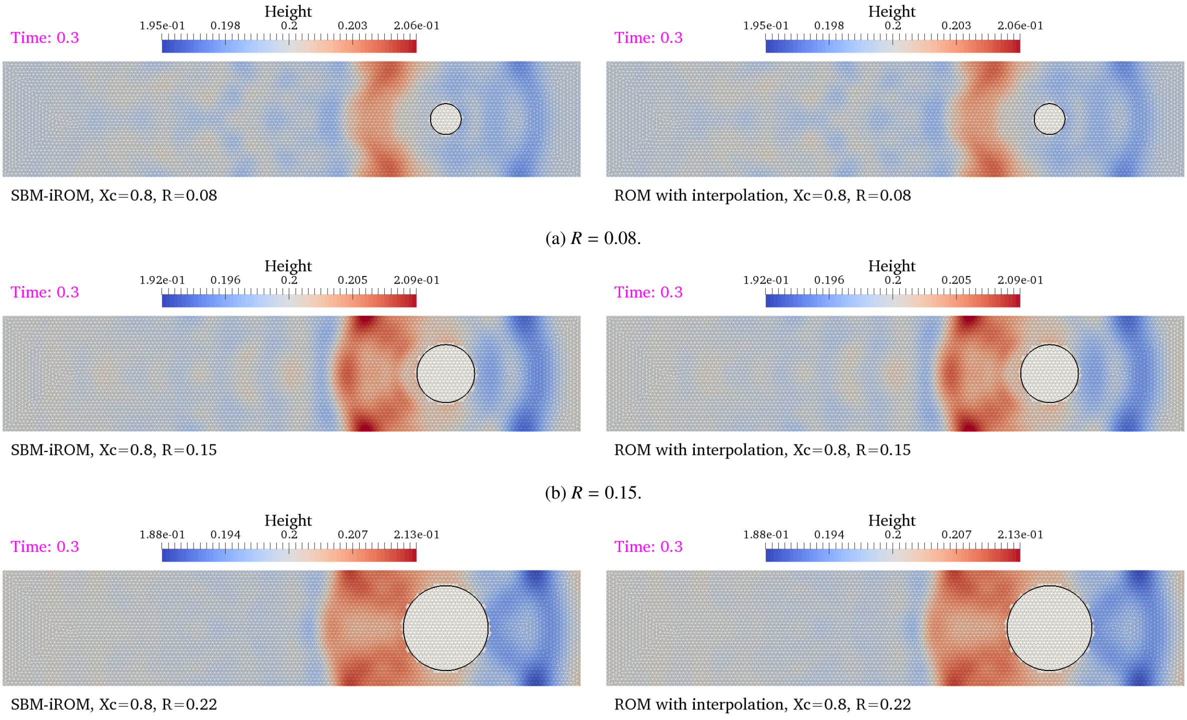
Figure 5.18: The height (h) in the case of $R = 0.08$ (top row), $R = 0.15$ (middle row), or $R = 0.22$ (bottom row), and $x_c = 0.3$ computed by FOM (left panels) and SBM-iROM (right panels), with the legend range set according to the FOM computations. The ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

Table 15: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.15$ and $x_c = -0.15$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.77e-3	2.53e-2	1.40e-1	3.26e-3	5.45e-2	2.99e-1	1.21e-2	1.55e-1	6.19e-1
$1 - 10^{-6}$	1.15e-3	1.64e-2	1.13e-1	1.58e-3	3.19e-2	1.73e-1	1.12e-2	1.21e-1	6.49e-1
$1 - 10^{-7}$	6.90e-4	7.94e-3	6.54e-2	1.05e-3	2.59e-2	1.41e-1	8.07e-3	8.99e-2	4.81e-1
$1 - 10^{-8}$	3.46e-4	4.74e-3	3.38e-2	5.21e-4	1.36e-2	7.70e-2	7.07e-3	8.09e-2	4.03e-1
$1 - 10^{-9}$	2.71e-4	3.77e-3	2.68e-2	4.34e-4	1.12e-2	6.19e-2	6.19e-3	7.14e-2	3.46e-1

Table 16: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.22$ and $x_c = -0.15$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.23e-3	2.72e-2	1.09e-1	3.44e-3	5.10e-2	2.37e-1	1.96e-2	2.46e-1	7.27e-1
$1 - 10^{-6}$	1.42e-3	1.94e-2	8.15e-2	2.23e-3	3.29e-2	1.35e-1	1.62e-2	1.89e-1	6.91e-1
$1 - 10^{-7}$	6.86e-4	8.64e-3	4.42e-2	1.16e-3	2.10e-2	9.41e-2	1.43e-2	1.71e-1	6.22e-1
$1 - 10^{-8}$	3.27e-4	5.09e-3	2.49e-2	5.72e-4	1.15e-2	5.33e-2	1.19e-2	1.43e-1	5.20e-1
$1 - 10^{-9}$	2.24e-4	3.98e-3	1.95e-2	4.49e-4	9.05e-3	3.78e-2	1.00e-2	1.23e-1	4.40e-1



(a) $R = 0.08$.

(b) $R = 0.15$.

(c) $R = 0.22$.

Figure 5.19: The height (h) in the case of $R = 0.08$ (top row), $R = 0.15$ (middle row), or $R = 0.22$ (bottom row), and $x_c = 0.8$ computed by FOM (left panels) and SBM-iROM (right panels), with the legend range set according to the FOM computations. The ROM computations are performed using POD modes corresponding to $\mathcal{E}_{\text{pod}} = 1 - 10^{-6}$.

Table 17: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.08$ and $x_c = 0.3$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	1.89e-3	4.15e-2	5.38e-1	2.96e-3	6.30e-2	7.55e-1	5.34e-2	2.79e-1	2.13e+0
$1 - 10^{-6}$	1.78e-3	3.43e-2	4.99e-1	2.26e-3	4.67e-2	5.94e-1	4.90e-2	3.17e-1	2.42e+0
$1 - 10^{-7}$	1.55e-3	2.48e-2	3.59e-1	1.68e-3	4.43e-2	4.89e-1	3.18e-2	4.06e-1	5.51e+0
$1 - 10^{-8}$	1.04e-3	1.67e-2	1.85e-1	9.46e-4	3.52e-2	3.23e-1	2.31e-2	2.98e-1	2.50e+0
$1 - 10^{-9}$	8.60e-4	1.33e-2	1.42e-1	6.98e-4	2.95e-2	2.80e-1	2.18e-2	2.92e-1	2.29e+0

Table 18: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.15$ and $x_c = 0.3$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1 - 10^{-5}$	2.24e-3	3.76e-2	2.13e-1	3.34e-3	5.79e-2	3.86e-1	1.28e-2	1.68e-1	7.46e-1
$1 - 10^{-6}$	1.88e-3	2.88e-2	1.87e-1	2.47e-3	4.78e-2	2.81e-1	1.32e-2	1.52e-1	7.98e-1
$1 - 10^{-7}$	1.20e-3	1.38e-2	1.35e-1	1.51e-3	3.58e-2	2.25e-1	9.16e-3	1.04e-1	6.12e-1
$1 - 10^{-8}$	6.31e-4	9.09e-3	7.39e-2	8.05e-4	2.43e-2	1.57e-1	6.38e-3	7.33e-2	4.45e-1
$1 - 10^{-9}$	5.23e-4	7.44e-3	5.71e-2	7.21e-4	1.98e-2	1.30e-1	5.30e-3	6.29e-2	3.71e-1

Table 19: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.22$ and $x_c = 0.3$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1-10^{-5}$	2.61e-3	3.66e-2	1.44e-1	4.12e-3	6.08e-2	2.72e-1	2.14e-2	2.54e-1	8.82e-1
$1-10^{-6}$	2.02e-3	2.72e-2	1.13e-1	2.86e-3	4.45e-2	2.09e-1	1.74e-2	2.06e-1	7.61e-1
$1-10^{-7}$	9.48e-4	1.03e-2	6.58e-2	1.67e-3	2.29e-2	1.14e-1	1.52e-2	1.83e-1	7.02e-1
$1-10^{-8}$	4.93e-4	6.32e-3	3.85e-2	8.07e-4	1.36e-2	7.29e-2	1.07e-2	1.30e-1	5.20e-1
$1-10^{-9}$	3.85e-4	5.07e-3	2.85e-2	6.24e-4	1.23e-2	6.33e-2	8.90e-3	1.11e-1	4.25e-1

Table 20: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.08$ and $x_c = 0.8$ of Test 3.

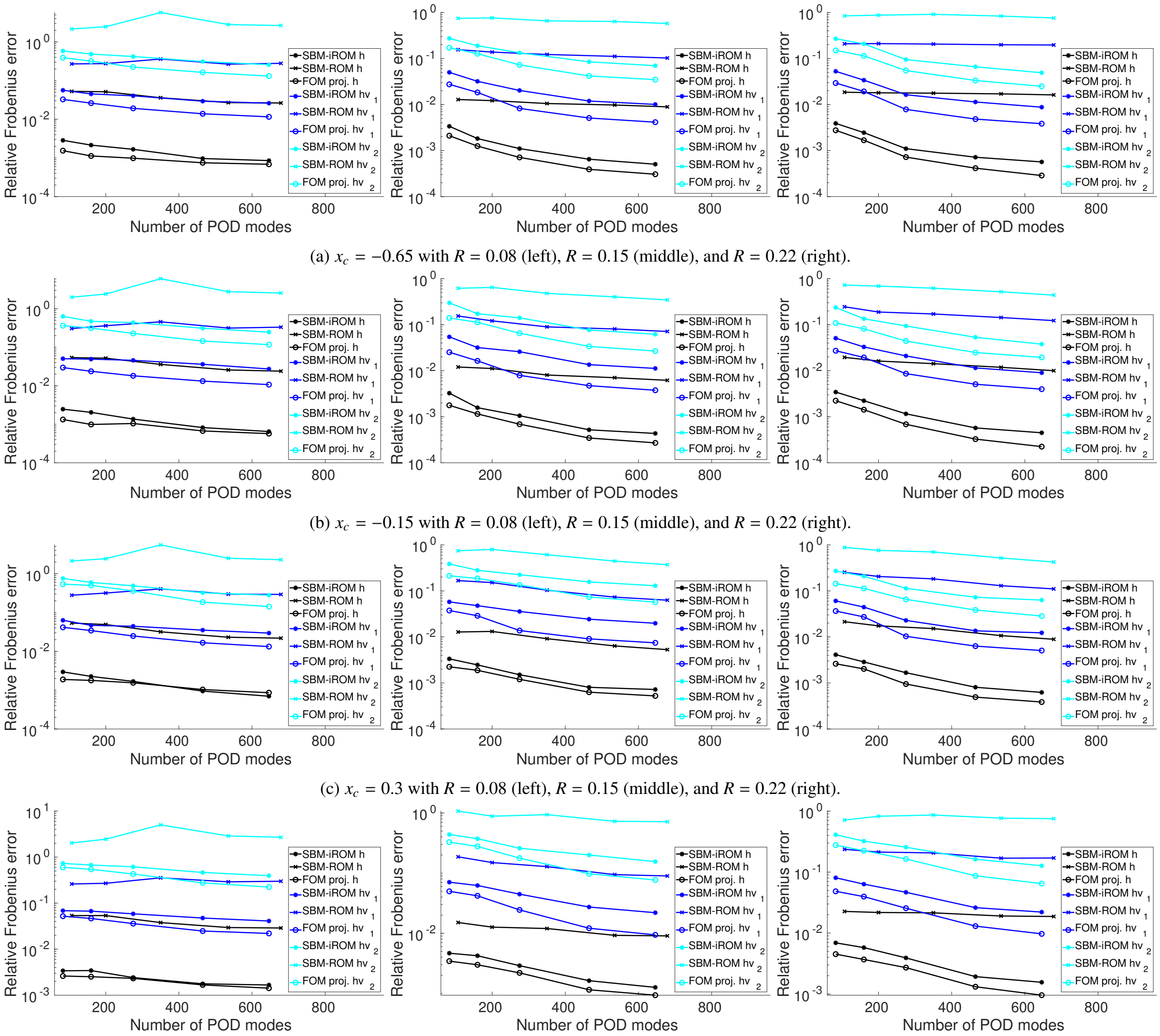
	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1-10^{-5}$	2.61e-3	5.21e-2	5.99e-1	3.40e-3	6.85e-2	7.29e-1	5.36e-2	2.61e-1	2.03e+0
$1-10^{-6}$	2.52e-3	4.65e-2	5.43e-1	3.43e-3	6.70e-2	6.73e-1	5.37e-2	2.70e-1	2.45e+0
$1-10^{-7}$	2.32e-3	3.59e-2	4.30e-1	2.41e-3	5.86e-2	6.16e-1	3.78e-2	3.55e-1	5.02e+0
$1-10^{-8}$	1.67e-3	2.48e-2	2.75e-1	1.77e-3	4.75e-2	4.64e-1	2.95e-2	2.91e-1	2.89e+0
$1-10^{-9}$	1.42e-3	2.20e-2	2.24e-1	1.66e-3	4.09e-2	3.96e-1	2.88e-2	2.99e-1	2.70e+0

Table 21: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.15$ and $x_c = 0.8$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1-10^{-5}$	3.45e-3	4.98e-2	3.28e-1	4.67e-3	7.07e-2	4.39e-1	1.51e-2	1.87e-1	1.07e+0
$1-10^{-6}$	2.99e-3	4.20e-2	2.78e-1	4.23e-3	6.23e-2	3.74e-1	1.27e-2	1.50e-1	8.85e-1
$1-10^{-7}$	2.20e-3	2.45e-2	1.78e-1	2.89e-3	4.47e-2	2.60e-1	1.20e-2	1.28e-1	9.38e-1
$1-10^{-8}$	1.15e-3	1.21e-2	9.76e-2	1.63e-3	2.73e-2	2.00e-1	9.24e-3	9.44e-2	7.28e-1
$1-10^{-9}$	9.35e-4	9.40e-3	7.69e-2	1.26e-3	2.20e-2	1.55e-1	9.06e-3	8.98e-2	7.16e-1

Table 22: The relative space-time Frobenius errors of the projected FOM solutions and ROM computations in the case $R = 0.22$ and $x_c = 0.8$ of Test 3.

	FOM projection			SBM-iROM			SBM-ROM		
$\mathcal{E}_{\text{pod}}$	h	hv_1	hv_2	h	hv_1	hv_2	h	hv_1	hv_2
$1-10^{-5}$	4.52e-3	4.84e-2	2.77e-1	6.92e-3	8.07e-2	4.10e-1	2.27e-2	2.36e-1	7.13e-1
$1-10^{-6}$	3.69e-3	3.96e-2	2.24e-1	5.78e-3	6.34e-2	3.22e-1	2.18e-2	2.13e-1	8.23e-1
$1-10^{-7}$	2.71e-3	2.56e-2	1.64e-1	3.92e-3	4.67e-2	2.56e-1	2.15e-2	2.06e-1	8.63e-1
$1-10^{-8}$	1.32e-3	1.30e-2	8.69e-2	1.93e-3	2.61e-2	1.63e-1	1.91e-2	1.69e-1	7.66e-1
$1-10^{-9}$	9.58e-4	9.72e-3	6.48e-2	1.56e-3	2.21e-2	1.27e-1	1.87e-2	1.70e-1	7.52e-1



(a) $x_c = -0.65$ with $R = 0.08$ (left), $R = 0.15$ (middle), and $R = 0.22$ (right).

(b) $x_c = -0.15$ with $R = 0.08$ (left), $R = 0.15$ (middle), and $R = 0.22$ (right).

(c) $x_c = 0.3$ with $R = 0.08$ (left), $R = 0.15$ (middle), and $R = 0.22$ (right).

(d) $x_c = 0.8$ with $R = 0.08$ (left), $R = 0.15$ (middle), and $R = 0.22$ (right).

Figure 5.20: The relative Frobenius errors of Test 3 by SBM-iROM (marked by $*$), SBM-ROM (marked by x), and FOM projection error (FOM proj., marked by o) in logarithmic scale plotted against the number of POD modes. The errors in h, hv_1, hv_2 are plotted in black, blue, and cyan colors, respectively.