Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 380 (2021) 113762
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Model order reduction of flow based on a modular geometrical approximation of blood vessels

Luca Pegolotti[a],[*], Martin R. Pfaller[b], Alison L. Marsden[b], Simone Deparis[a]

[a] SCI-SB-SD, Institute of Mathematics, École Polytechnique Fédérale de Lausanne, Station 8, EPFL, CH–1015 Lausanne, Switzerland
[b] Department of Pediatrics (Cardiology), Bioengineering, Stanford University, Clark Center E1.3, 318 Campus Drive, Stanford, CA 94305, USA

Received 5 October 2020; received in revised form 3 March 2021; accepted 3 March 2021
Available online xxxx

Abstract

We are interested in a reduced order method for the efficient simulation of blood flow in arteries. The blood dynamics is modeled by means of the incompressible Navier–Stokes equations. Our algorithm is based on an approximated domain-decomposition of the target geometry into a number of subdomains obtained from the parametrized deformation of geometrical building blocks (e.g., straight tubes and model bifurcations). On each of these building blocks, we build a set of spectral functions by Proper Orthogonal Decomposition of a large number of snapshots of finite element solutions (offline phase). The global solution of the Navier–Stokes equations on a target geometry is then found by coupling linear combinations of these local basis functions by means of spectral Lagrange multipliers (online phase). Being that the number of reduced degrees of freedom is considerably smaller than their finite element counterpart, this approach allows us to significantly decrease the size of the linear system to be solved in each iteration of the Newton–Raphson algorithm. We achieve large speedups with respect to the full order simulation (in our numerical experiments, the gain is at least of one order of magnitude and grows inversely with respect to the reduced basis size), whilst still retaining satisfactory accuracy for most cardiovascular simulations.

Keywords: Cardiovascular simulations; Model order reduction; Reduced basis method; Domain-decomposition

1. Introduction

Cardiovascular disease is the leading cause of death worldwide. This term broadly encompasses a variety of pathological cases ranging from heart disease to many other peripheral vascular diseases. Consequently, the numerical simulation of blood flow in the cardiovascular system has gained considerable attention during the last twenty years as a valuable quantitative tool for the study and diagnosis of such conditions [1,2].

Blood dynamics is typically modeled by means of the incompressible Navier–Stokes equations; their discretization by numerical methods such as the finite element (FE) method leads to the *Full Order Model* (FOM). Despite the rapid and constant growth in computational power of the architectures that are currently employed to run such simulations, the time and resources required are often incompatible with clinical practice. Moreover, in many cases the numerical results of the FOM are affected by a level of uncertainty not inherently associated with the employed

[*] Corresponding author.
E-mail address: luca.pegolotti@epfl.ch (L. Pegolotti).

https://doi.org/10.1016/j.cma.2021.113762

numerical method but rather with the inexact geometry and boundary conditions considered in the simulation [3,4]. *Reduced Order Models* (ROMs) aim at lowering the computational burden of FOMs at the cost of settling for larger approximation errors. This is particularly desirable in multi-query scenarios, i.e., whenever the same simulation needs to be performed for multiple values of the input parameters (for instance, in order to quantify the uncertainty due to the problem data [5]).

Among the ROMs employed in the context of cardiovascular modeling are the popular 0D and 1D models [6–8]. These models consider a coarse approximation of systems of arteries as electric circuits (0D) or as segments in which the quantities of interest are obtained by an averaging process across the section of the vessels (1D). They often prove to be remarkably accurate in approximating flow rates and pressure drops [9], albeit the strong geometrical approximations inevitably entail a significant loss of local details. For this reason, algorithms to couple 0D and 1D models with full 3D simulations—to be employed in the regions in which higher quality solutions are required—have been devised [10–12].

Some recent works have introduced strategies to overcome the main limitation of 1D models, namely that only the axial component of the flow is accurately characterized. In particular, in [13,14], the authors propose considering different discretization methods for the longitudinal and transversal directions (for example, the finite element method in the former and spectral methods in the latter). The Transversally Enriched Pipe Element Method (TEPEM) [15,16] further develops this idea: here, the geometry is decomposed into several pipe-type elements in which the two flow components are separately discretized. This method has also been applied to uncertainty quantification [17].

In this paper, we aim at formalizing a ROM allowing to approximate the local features of the blood flow. The strategy is based on the combination of a domain-decomposition approach with the Reduced Basis (RB) method and can be interpreted as a specific implementation of the Reduced Basis Element (RBE) method [18,19]. We refer the reader to [20–24] for uses of the RBE method for the approximation of the 2D steady Stokes equations in the context of the cardiovascular system. To our knowledge, our work represents the first application of the RBE method to the unsteady 3D Navier–Stokes equations.

Similarly to 1D models, the proposed method is based on the approximation of the vessel geometry as a composition of simple subdomains. However, in our approach, these subdomains are three dimensional and obtained from the parametrized geometrical deformation of a handful of elementary building blocks (e.g., straight tubes and bifurcations). Each building block in its reference configuration is equipped with a set of spectral basis functions. Since in this paper we focus on the FE method to generate the FOMs, these are FE functions defined over triangulations of the building block. Specifically, the basis functions are found by means of Proper Orthogonal Decomposition (POD) of a large number of flow solutions which are computed during a computationally expensive offline data generation phase. The global flow approximations by our ROM is computed as a composition of local (to the subdomains) solutions, namely linear combinations of the basis functions defined in every subdomain scaled by the divergence-free Piola transformation. The local solutions are coupled by a nonconforming domain-decomposition method based on the use of spectral Lagrange multipliers on the 2D interfaces [25].

The gain in performance with respect to the FOM is given by the decreased size of the linear system to be solved in each iteration of the Newton–Raphson algorithm. Indeed, while the number of degrees of freedom in the FE model is typically large (tens or hundreds of thousands per each subdomain in our numerical simulations), only a few hundred basis functions per subdomain are sufficient to retain acceptable levels of accuracy.

The rest of this paper is structured as follows. In Section 2, we provide a self-contained and concise introduction to the RB method in order to set the notation and terminology for the remaining sections. Section 3 is dedicated to the Navier–Stokes equations and their numerical discretization by the FE method. In Section 4, we define the concept of modular approximation of arteries by the above-mentioned domain-decomposition approach; we also address the numerical solution of the Navier–Stokes equations on the decomposed geometries by the FE method and the nonconforming domain-decomposition method. In particular, we devise an ad-hoc preconditioner that takes advantage of the peculiar block structure of the global system matrix. It is worth noticing that, although the paper focuses on a ROM, addressing the solution of the partitioned problem with the FE method is necessary, as the RB functions in the subdomains are generated by POD of local solutions obtained from global problems in decomposed domains. This strategy of data collection (offline phase) is discussed in Section 5. In the same section, we also delineate the algorithm for the approximation of the global solution on a decomposed target geometry using the ROM (online phase). Our numerical results are reported in Section 6. Specifically, Sections 6.1 and 6.2, respectively, focus on two critical points: (i) the accuracy with respect to the corresponding FE solution in the decomposed geometry—which, in our case, plays the role of FOM—and the achieved speedup, and (ii) the comparison of the reduced solution to the one obtained on a physiological and non-decomposed geometry. In the latter case, we mostly aim at evaluating the effects of the geometrical approximation on the local features of the flow, for instance, in terms of the wall-shear stress (WSS). Finally, in Section 7 we draw our conclusions and discuss future perspectives of the current study.

1.1. Notation

Vectors and matrices. We use the notation $\mathbf{a} : \Omega \mapsto \mathbb{R}^d$ to indicate d-dimensional vector fields defined over some domain $\Omega \subset \mathbb{R}^d$, whereas $\underline{\mathbf{a}} \in \mathbb{R}^m$ refers to an algebraic column vector. When considering multiple subdomains, local matrices and global block matrices are written as A and $\mathcal{A}$, respectively. Global block algebraic vectors are written in capital letters and we use the notation $\underline{\mathbf{A}} = \text{vec}(\underline{\mathbf{a}}_1, \dots, \underline{\mathbf{a}}_n) \in \mathbb{R}^M$, $M = \sum_{i=1}^n m_i$, to indicate the concatenation of $\underline{\mathbf{a}}_1 \in \mathbb{R}^{m_1}, \dots, \underline{\mathbf{a}}_n \in \mathbb{R}^{m_n}$. On the other hand, the notation $A = [\underline{\mathbf{a}}_1 | \dots | \underline{\mathbf{a}}_n] \in \mathbb{R}^{m \times n}$ indicates a matrix whose columns are represented by $\underline{\mathbf{a}}_1 \in \mathbb{R}^m, \dots, \underline{\mathbf{a}}_n \in \mathbb{R}^m$.

Superscripts, subscripts and hats. We use the superscripts h or N to denote quantities related to the FE or the RB approximations. In Section 4, we introduce a modular domain-decomposition method based on reference building blocks and subdomains of the target geometry. Indices of the subdomains are written as simple superscripts (e.g., Ω^j); superscripts in square brackets refer to interfaces (e.g., $\Gamma^{[ij]} = \overline{\Omega}^i \cap \overline{\Omega}^j$) or portions of boundaries and related quantities. In the context of the Newton–Raphson algorithm, superscripts in parentheses are used to refer to the iteration index of the iterative method. Subscripts are used to indicate: (i) indices of vectors or components (or blocks) of vectors or matrices, and (ii) quantities at specific timesteps (e.g., $\underline{\mathbf{u}}_i^h$ is the vector of FE degrees of freedom of velocity at the ith timestep). Whenever a symbol refers to a reference building block or the reference interface (unit disk), it is indicated with a hat notation (e.g., $\widehat{\Omega}^i$ is the ith reference building block).

2. The reduced basis method in a nutshell

In this section, we provide a non-comprehensive introduction to the RB method which is intended to set the theoretical basis for the remainder of the paper. For a complete overview, we refer the reader to [26,27].

Let us consider an open and bounded domain Ω and a steady differential problem of the form

$$\mathscr{L}(u; \underline{\boldsymbol{\mu}}) = \mathscr{G}(\underline{\boldsymbol{\mu}}), \tag{1}$$

where $u \in \mathcal{V}$ ($\mathcal{V}$ being a suitable functional space) is the solution, $\underline{\boldsymbol{\mu}} \in \mathcal{D} \subset \mathbb{R}^{N_\mu}$ is a vector of geometrical and/or physical parameters, $\mathscr{L}$ is a generic differential operator, and $\mathscr{G}$ is a functional encoding the data of the problem, such as forcing term and boundary conditions. In this section we assume that $\mathscr{L}$ is an elliptic operator. The extension of this setting to the Navier–Stokes equations is considered in Section 5.

The standard approach to solve Eq. (1) by a Galerkin method corresponds to transforming the continuous problem into a finite dimensional one, often referred to as *Full Order Model* (FOM). For instance, in Section 3.2 we discuss how the numerical solution of the Navier–Stokes equations by the FE method is found as a linear combination of the FE basis functions. With respect to the model problem in Eq. (1), this translates to $u^h = \sum_{i=1}^{N^h} u_i^h \varphi_i^h$, where $\varphi_i^h \in \mathcal{V}^h \subset \mathcal{V}$ are FE basis functions; $\underline{\mathbf{u}}^h = [u_1^h, \dots, u_{N^h}^h]$ is typically called vector of degrees of freedom. Assuming that the differential operator $\mathscr{L}$ can be mathematically described in the weak sense by a bilinear form as $\ell(\varphi, \psi)$, for $\varphi \in \mathcal{V}$ and $\psi \in \mathcal{V}$, we identify the matrix $L^h(\underline{\boldsymbol{\mu}})_{ij} = \ell(\varphi_j^h, \varphi_i^h; \underline{\boldsymbol{\mu}}) \in \mathbb{R}^{N^h \times N^h}$, and similarly $\underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}})_i = \int_\Omega \mathscr{G}(\underline{\boldsymbol{\mu}}) \varphi_i^h \in \mathbb{R}^{N^h}$. For example, for the linear differential operator $\mathscr{L}(u; \eta) = \eta \Delta u$, describing a Poisson equation with parameter η, we have $\ell(\varphi_j^h, \varphi_i^h; \eta) = \int_\Omega \eta \nabla \varphi_j^h \cdot \nabla \varphi_i^h$. The resulting linear system of dimension $N^h \times N^h$

$$L^h(\underline{\boldsymbol{\mu}}) \underline{\mathbf{u}}^h = \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}}) \tag{2}$$

is possibly very large and expensive to solve. In multi-query scenarios—i.e., whenever it is required to solve Eq. (2) for multiple values of the parameter $\underline{\boldsymbol{\mu}}$—it is often crucial to reduce the dimensionality of the system in order to save computational time. One way to achieve this is by employing ROMs, such as the RB method.

The main idea of the RB method is to construct a low dimensional basis for the solution u out of a number of solutions N_s (snapshots) of the FOM, which are computed during the so-called *offline phase*. In the *online phase*, the reduced solution is obtained as a linear combination of the RB functions; system (2) is cast in the form of a small linear system where the unknowns represent the coefficients of such a linear combination. In the remainder of this section, we address the offline and online phases more in depth.

2.1. The offline phase: basis construction

There exist two main strategies for the construction of the reduced basis: greedy algorithms [28,29] and the Proper Orthogonal Decomposition (POD) method. The former lead to a more efficient offline phase, as they allow to minimize the number of snapshots N_s to be computed. A major drawback of greedy algorithms is that they are based on an a posteriori estimate of the projection error, which is often difficult to compute in practical applications. For this reason, in this paper we opt for the POD method, which often requires a larger number of snapshots N_s but is in turn more general. We refer, e.g., to [30,31] and [32] for applications of POD to parabolic and fluid problems, respectively, and [31] for [30] for a comprehensive study of the properties of POD when applied to the solution of Ordinary Differential Equations (ODEs). In the context of cardiac simulations, this technique has been successfully employed both in fluid (e.g., in [33] to simulate blood flow in patient-specific coronary artery bypass grafts) and structural simulations (e.g., in [34], where POD is used to reduce the space of admissible displacements of the heart muscle). In the POD approach, the reduced basis is usually constructed by singular value decomposition (SVD) [35,36] out of the set of snapshots, which are obtained by sampling N_s parameters $\underline{\boldsymbol{\mu}}_1, \dots, \underline{\boldsymbol{\mu}}_{N_s}$ in $\mathcal{D}$ and by solving the corresponding FOM. Formally, we arrange the snapshots in matrix form as $S = [\underline{\mathbf{u}}_1^h | \dots | \underline{\mathbf{u}}_{N_s}^h] \in \mathbb{R}^{N^h \times N_s}$ and we seek matrices $U = [\underline{\boldsymbol{\zeta}}_1^h | \dots | \underline{\boldsymbol{\zeta}}_{N_s}^h] \in \mathbb{R}^{N^h \times N_s}$, $\Sigma \in \mathbb{R}^{N_s \times N_s}$ and $Z \in \mathbb{R}^{N_s \times N_s}$ such that $S = U \Sigma Z^{\mathrm{T}}$; the columns of U and Z are orthonormal. In the context of POD, $\underline{\boldsymbol{\zeta}}_1^h, \dots, \underline{\boldsymbol{\zeta}}_{N_s}^h$ are often called modes. We remark that in classic SVD the matrices U and Σ are of size $N^h \times N^h$ and $N^h \times N_s$, respectively; here we consider the "economic" version of the algorithm. Matrix Σ takes the form $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_{N_s})$ and its diagonal is composed of the singular values of matrix S ordered from largest to smallest, i.e., $\sigma_1 \geq \dots \geq \sigma_{N_s} \geq 0$. Let us define $V := [\underline{\boldsymbol{\zeta}}_1^h | \dots | \underline{\boldsymbol{\zeta}}_N^h] \in \mathbb{R}^{N^h \times N}$ as the matrix composed of the first N modes and let us recall that, given a N-dimensional orthonormal basis, arranged in matrix form as $W = [\underline{\mathbf{w}}_1 | \dots | \underline{\mathbf{w}}_N] \in \mathbb{R}^{N^h \times N}$, the projection of a generic vector $\underline{\mathbf{x}} \in \mathbb{R}^{N^h}$ onto span$\{\underline{\mathbf{w}}_1 | \dots | \underline{\mathbf{w}}_N\}$ is given by $\Pi_W \underline{\mathbf{x}} = W W^{\mathrm{T}} \underline{\mathbf{x}}$. Then, the following proposition holds.

Proposition 1. *Let $\mathcal{V}_N = \{W \in \mathbb{R}^{N^h \times N} : W^T W = I\}$ be the set of all matrices whose columns form a N-dimensional orthonormal basis. Then,*

$$\sum_{i=1}^{N_s} \|\underline{\mathbf{u}}_i - V V^T \underline{\mathbf{u}}_i\|_2^2 = \min_{W \in \mathcal{V}_N} \sum_{i=1}^{N_s} \|\underline{\mathbf{u}}_i - W W^T \underline{\mathbf{u}}_i\|_2^2 = \sum_{i=N+1}^{N_s} \sigma_i^2.$$

We refer to [26] for a proof of Proposition 1. In other words, the columns of V are the N-dimensional basis minimizing the projection error of the snapshots over its column space; moreover, such error is strictly related to the magnitude of the singular values $\sigma_{N+1}, \dots, \sigma_{N_s}$. Thus, a common heuristic to choose N is to set it equal to the smallest integer $\widetilde{N}$ such that

$$\frac{\sum_{i=1}^{\widetilde{N}} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \geq 1 - \varepsilon^2, \tag{3}$$

where ε is a user-provided tolerance. The left hand side of Eq. (3) is the relative information content of the POD basis, namely the percentage of energy of the snapshots retained by the first $\widetilde{N}$ modes. The size of the reduced basis N selected by following criterion (3) is typically much smaller than the size of the FOM N^h, i.e., $N \ll N^h$.

Remark 1. Given a symmetric positive definite matrix X^h which is a norm matrix for $\|\cdot\|_{\mathcal{V}}$ in the FE space, i.e., $\|u\|_{\mathcal{V}} = (\underline{\mathbf{u}}^h)^{\mathrm{T}} X^h \underline{\mathbf{u}}^h$, it is possible to perform the POD such that the basis U is orthonormal with respect to X^h (i.e., $U^{\mathrm{T}} X^h U = I$). In order to achieve this, we observe that, since X^h is symmetric positive definite, it admits a Cholesky decomposition $X^h = H^{\mathrm{T}} H$, H being upper triangular. Matrix U is then found as $U = H^{-1} \widetilde{U}$, where $\widetilde{U}$ is computed by SVD of $HS = \widetilde{U} \widetilde{\Sigma} \widetilde{V}^{\mathrm{T}}$. When constructing the reduced basis for our particular application in

Section 5, following this approach allows us to achieve the optimality expressed in Proposition 1 with respect to norms more suited to the specific variables of interest (namely, H^1 norm for the velocity and L^2 norm for the pressure).

2.2. The online phase: solution of the reduced problem

Let us observe that it is legitimate to associate with each POD mode a corresponding functional representation $\zeta_j^h = \sum_{i=1}^{N^h} (\underline{\boldsymbol{\zeta}}_j^h)_i \varphi_i^h$. The RB approximation then reads $u^N = \sum_{i=1}^{N} u_i^N \zeta_i^h$, $\underline{\mathbf{u}}^N = [u_1^N, \dots, u_N^N]$ being the vector of reduced degrees of freedom. Then, evaluating the weak formulation of problem (1) at test and trial functions in $\text{span}\{\zeta_i^h\}_{i=1}^N$, we find the reduced linear system

$$L^N(\underline{\boldsymbol{\mu}})\underline{\mathbf{u}}^N = \underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}), \tag{4}$$

where $L^N(\underline{\boldsymbol{\mu}})_{ij} = \ell(\zeta_j^h, \zeta_i^h) \in \mathbb{R}^{N \times N}$ and $\underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}})_i = \int_\Omega \mathscr{G}(\underline{\boldsymbol{\mu}}) \zeta_i^h \in \mathbb{R}^N$. The assembly and solution of system (4) correspond to the online phase. The transformation of the reduced vector of degrees of freedom into its FE counterpart is simply performed by $\underline{\mathbf{u}}^h \approx V \underline{\mathbf{u}}^N \in \mathbb{R}^{N^h}$.

By exploiting the expansion $\zeta_j^h = \sum_{i=1}^{N^h} (\underline{\boldsymbol{\zeta}}_j^h)_i \varphi_i^h$ it is easy to find that $L^N(\underline{\boldsymbol{\mu}}) = V^{\mathrm{T}} L^h(\underline{\boldsymbol{\mu}}) V$ and $\underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}) = V^{\mathrm{T}} \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}})$. In the most general case, therefore, the assembly of the reduced system is done by constructing the full order matrix and right hand side and by computing their projection onto the RB space. If the problem features an affine decomposition, namely there exist parameter-dependent coefficients θ_q^L for $q = 1, \dots, Q_L$ and θ_q^G for $q = 1, \dots, Q_G$ such that

$$L^h(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_L} \theta_q^L(\underline{\boldsymbol{\mu}}) L_q^h, \qquad \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_G} \theta_q^G(\underline{\boldsymbol{\mu}}) \underline{\mathbf{G}}_q^h, \tag{5}$$

a considerable speedup is achieved by precomputing the matrices $L_q^N = V^{\mathrm{T}} L_q^h V$ and the vectors $\underline{\mathbf{G}}_q^N = V^{\mathrm{T}} \underline{\mathbf{G}}_q^h$ in the offline phase, and by assembling the reduced elements of system (4) as

$$L^N(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_L} \theta_q^L(\underline{\boldsymbol{\mu}}) L_q^N, \qquad \underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_G} \theta_q^G(\underline{\boldsymbol{\mu}}) \underline{\mathbf{G}}_q^N.$$

Unfortunately, in most practical scenarios an affine decomposition of the form (5) is not readily available. In such cases, a common strategy to efficiently perform the assembly of system (4) consists in employing the (discrete) empirical interpolation method (DEIM) [37,38] and its matrix variant MDEIM [39]. In this paper, we do not address the optimization of the assembly of the reduced system, which will be investigated in future works.

Remark 2. The quality of the RB approximation depends on three factors: the POD tolerance ε, the number of considered snapshots N_s, and the choice of sampling space. Assuming the latter to be appropriate, however, the number of snapshots required to achieve errors of the order of the POD tolerance in the online phase may become too large as the dimension of the parameter space $\mathcal{D}$ increases. In other words, in applications were the space of parameters is too rich, it is unfeasible to sample a sufficient number of snapshots, and the online error of the RB method might be considerably greater than the one obtained on the snapshots.

3. The Navier–Stokes equations

In this section, we first introduce the Navier–Stokes equations in strong and weak formulations (Section 3.1). The latter poses the mathematical foundation for the numerical discretization in space by the FE method as described in Section 3.2, where we also derive the fully-discrete model by considering a generic Backward Differentiation Formulas (BDF) scheme.

3.1. Strong and weak formulations

Let us consider the problem of approximating the blood flow in a vessel, mathematically represented by an open and bounded domain $\Omega \in \mathbb{R}^d$. In this paper, we take $d = 3$, but the discussion is also valid for the case

$d = 2$. We model the blood as an incompressible Newtonian fluid and, therefore, its dynamics is described by the Navier–Stokes equations

$$\begin{aligned} \rho_{\mathrm{f}}\frac{\partial \mathbf{u}}{\partial t} + \rho_{\mathrm{f}}(\mathbf{u}\cdot\nabla)\mathbf{u} - 2\mu_{\mathrm{f}}\nabla\cdot\varepsilon(\mathbf{u}) + \nabla p = \mathbf{f} \qquad & \text{in } \Omega\times(0,T), \\ \nabla\cdot\mathbf{u} = 0 \qquad & \text{in } \Omega\times(0,T), \\ \mathbf{u} = \mathbf{g} \qquad & \text{on } \Gamma_D\times(0,T), \\ \sigma(\mathbf{u},p)\mathbf{n} = \mathbf{h} \qquad & \text{on } \Gamma_N\times(0,T), \\ \mathbf{u} = \mathbf{u}_0 \qquad & \text{for } t = 0, \end{aligned} \tag{6}$$

where $\mathbf{u} : \Omega\times(0,T)\mapsto\mathbb{R}^d$ and $p : \Omega\times(0,T)\mapsto\mathbb{R}$ are velocity and pressure of the fluid, ρ_{f} is the density, μ_{f} is the viscosity, $\varepsilon(\mathbf{u}) = (\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})/2$ is the strain rate tensor, $\sigma(\mathbf{u},p) = 2\mu_{\mathrm{f}}\varepsilon(\mathbf{u}) - pI$ is the Cauchy stress tensor, $\mathbf{f} : \Omega\times(0,T)\mapsto\mathbb{R}^d$ is a forcing term, $\mathbf{g} : \Gamma_D\times(0,T)\mapsto\mathbb{R}^d$ and $\mathbf{h} : \Gamma_N\times(0,T)\mapsto\mathbb{R}^d$ are Dirichlet and Neumann data, $\mathbf{n}$ is the normal unit vector to the boundary $\partial\Omega$, and $\mathbf{u}_0 : \Omega\mapsto\mathbb{R}^d$ is the prescribed initial condition. Since we deal with cardiovascular applications, we take $\Gamma_D = \Gamma_{\mathrm{w}}\cup\left(\bigcup_{i=1}^{N_{\mathrm{in}}}\Gamma_{\mathrm{in}}^{[i]}\right)$ and $\Gamma_N = \bigcup_{i=1}^{N_{\mathrm{out}}}\Gamma_{\mathrm{out}}^{[i]}$; $\Gamma_{\mathrm{in}}^{[1]},\ldots,\Gamma_{\mathrm{in}}^{[N_{\mathrm{in}}]}$, Γ_{w} and $\Gamma_N^{[1]},\ldots,\Gamma_{\mathrm{out}}^{[N_{\mathrm{out}}]}$ are the inlets, wall and outlets of the vessel, respectively. The inlet velocity profiles and outlet Neumann data are denoted $\mathbf{g}_1,\ldots,\mathbf{g}_{N_{\mathrm{in}}}$ and $\mathbf{h}_1,\ldots,\mathbf{h}_{N_{\mathrm{out}}}$; on the wall Γ_{w} we consider $\mathbf{u} = \mathbf{0}$. The first equation in Eq. (6) (*momentum equation*) represents the generalization of Newton's second law of motion to continuums, and the second equation (*continuity equation*) is the incompressibility constraint.

The numerical solution of the Navier–Stokes equations by classical Galerkin methods such as the FE method entails transforming Eq. (6) into its weak formulation. Let us denote $\mathcal{V}_g := [H^1_{g,\Gamma_D}(\Omega)]^d$—that is, the space of functions belonging to $[H^1(\Omega)]^d$ such that their trace is equal to $\mathbf{g}$ on Γ_D, $\mathcal{Q} := L^2(\Omega)$, $\mathcal{V}_0 := [H^1_{0,\Gamma_D}(\Omega)]^d$, and let us consider generic test functions $\mathbf{v}\in\mathcal{V}_0$ and $q\in\mathcal{Q}$. The weak formulation of Eq. (6) is obtained by multiplying the momentum and continuity equations by $\mathbf{v}$ and q respectively and by integrating over the domain Ω. Hence, we find:

(W1) *given* $\mathbf{f}$, $\mathbf{g}$, *and* $\mathbf{h}$ *regular enough, find* $(\mathbf{u},p)\in\mathcal{V}_g\times\mathcal{Q}$, *such that, for every* $t\in(0,T)$,

$$\begin{aligned} \int_\Omega \rho_{\mathrm{f}}\frac{\partial\mathbf{u}}{\partial t}\cdot\mathbf{v} + \int_\Omega \rho_{\mathrm{f}}[(\mathbf{u}\cdot\nabla)\mathbf{u}]\cdot\mathbf{v} + \int_\Omega \sigma(\mathbf{u},p):\nabla\mathbf{v} = \int_\Omega \mathbf{f}\cdot\mathbf{v} + \int_{\Gamma_N}\mathbf{h}\cdot\mathbf{v} \qquad & \forall\mathbf{v}\in\mathcal{V}_0, \\ \int_\Omega \nabla\cdot\mathbf{u}q = 0 \qquad & \forall q\in\mathcal{Q}, \end{aligned}$$

and such that $\mathbf{u} = \mathbf{u}_0$ *for* $t = 0$.

3.2. Numerical discretization

In order to transform the infinite dimensional problem W1 into a finite dimensional one we consider the two subspaces $\mathcal{V}_g^h\subset\mathcal{V}_g$ and $\mathcal{Q}^h\subset\mathcal{Q}$. Then, we introduce the sets of basis functions $\boldsymbol{\varphi}_i^h$, $i = 1,\ldots,N_u^h$, and ψ_i^h, $i = 1,\ldots,N_p^h$, such that

$$\mathcal{V}_g^h = \mathrm{span}\{\boldsymbol{\varphi}_i^h\}_{i=1}^{N_u^h}, \qquad \mathcal{Q}^h = \mathrm{span}\{\psi_i^h\}_{i=1}^{N_p^h}.$$

The finite dimensional approximations of velocity and pressure read $\mathbf{u}^h(\mathbf{x},t) = \sum_{i=1}^{N_u^h} u_i^h(t)\boldsymbol{\varphi}_i^h(\mathbf{x})$ and $p^h(\mathbf{x},t) = \sum_{i=1}^{N_p^h} p_i^h(t)\psi_i^h(\mathbf{x})$, respectively; in these expressions we explicitly highlight for the sake of clarity the dependence of each term on space and time, but this is omitted hereon. The definition of $\mathcal{V}_g^h$ and $\mathcal{Q}^h$ clearly plays a crucial role in the accuracy of the approximation. Moreover, in the case of saddle-point problems such as the Navier–Stokes equations, the quality of the discretization is critical to ensure the well-posedness of the discrete problem; we refer the reader to Section 5.1 and [40,41] for details. In this section and in Section 4, the basis functions $\{\boldsymbol{\varphi}_i^h\}_{i=1}^{N_u^h}$ and $\{\psi_i^h\}_{i=1}^{N_p^h}$ are standard Lagrangian P2-P1 Taylor–Hood FE basis functions [42] (i.e., quadratic and linear piece-wise polynomials for the velocity and the pressure, respectively) obtained from a triangulation of the domain $\mathscr{T}^h$ composed of tetrahedra. We consider other possibilities—i.e., RB functions—in Section 5.

By introducing the vectors of degrees of freedom $\underline{\mathbf{u}}^h = [u_1^h, \dots, u_{N_u^h}^h] \in \mathbb{R}^{N_u^h}$, $\underline{\mathbf{p}}^h = [p_1^h, \dots, p_{N_p^h}^h] \in \mathbb{R}^{N_p^h}$ and $\underline{\mathbf{f}}_i^h = \int_\Omega \mathbf{f} \cdot \boldsymbol{\varphi}_i^h + \int_{\Gamma_N} \mathbf{h} \cdot \boldsymbol{\varphi}_i^h \in \mathbb{R}^{N_u^h}$, and the matrices $M_{ij}^h = \int_\Omega \mu_\mathrm{f} \boldsymbol{\varphi}_j^h \cdot \boldsymbol{\varphi}_i^h \in \mathbb{R}^{N_u^h \times N_u^h}$ (mass), $K_{ij}^h = \int_\Omega 2\mu_\mathrm{f} \varepsilon(\boldsymbol{\varphi}_j^h) : \varepsilon(\boldsymbol{\varphi}_i^h) \in \mathbb{R}^{N_u^h \times N_u^h}$ (stiffness), $C^h(\mathbf{u}^h)_{ij} = \int_\Omega \rho_\mathrm{f}[(\mathbf{u}^h \cdot \nabla)\boldsymbol{\varphi}_j^h] \cdot \boldsymbol{\varphi}_i^h \in \mathbb{R}^{N_u^h \times N_u^h}$ (convective matrix), and $D_{ij}^h = -\int_\Omega \nabla \cdot \boldsymbol{\varphi}_j^h \psi_i^h \in \mathbb{R}^{N_p^h \times N_u^h}$ (divergence), the discrete version of W1 is conveniently expressed in the form of linear system as

$$\begin{bmatrix} M^h & \\ & \end{bmatrix} \begin{bmatrix} \dot{\underline{\mathbf{u}}}^h \\ \dot{\underline{\mathbf{p}}}^h \end{bmatrix} + \begin{bmatrix} K^h + C^h\left(\mathbf{u}^h\right) & (D^h)^\mathrm{T} \\ D^h & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^h \\ \underline{\mathbf{p}}^h \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}}^h \\ \underline{\mathbf{0}} \end{bmatrix}. \tag{7}$$

We exploit that $\mathbf{u}^h = \sum_{i=1}^{N_u^h} u_i^h \boldsymbol{\varphi}_i^h$ and indicate the convective term matrix as $C^h(\underline{\mathbf{u}}^h)$—i.e., as a function of the degrees of freedom $\underline{\mathbf{u}}^h$ instead of the approximated function $\mathbf{u}^h$—in the remainder of the paper.

Let us now introduce a sequence of timesteps $t_0, t_1, \dots, t_{N_t}$ such that $t_0 = 0$, $t_{N_t} = T$, and $t_{k+1} = t_k + \Delta t$ for every $k = 0, \dots, N_t$; Δt is called timestep size. We denote the value of $\underline{\mathbf{u}}^h$ and $\underline{\mathbf{p}}^h$ at timestep t_k by $\underline{\mathbf{u}}^h(t_k) = \underline{\mathbf{u}}_k^h$ and $\underline{\mathbf{p}}^h(t_k) = \underline{\mathbf{p}}_k^h$, respectively. The numerical discretization in time of Eq. (7) is performed by means of BDF schemes. Specifically, given $\underline{\mathbf{u}}_{k-j+1}^h$ and $\underline{\mathbf{p}}_{k-j+1}^h$ for $j = 1, \dots, \sigma$, the numerical solution of the Navier–Stokes equations at timestep t_{k+1} by a BDF scheme of order σ satisfies

$$\underline{\mathbf{r}}\left(\underline{\mathbf{w}}_{k+1}^h\right) := H^h \underline{\mathbf{w}}_{k+1}^h - \sum_{j=1}^{\sigma} \alpha_j H^h \underline{\mathbf{w}}_{k-j+1}^h - \Delta t \beta \mathring{\underline{\mathbf{f}}}^h\left(t_{k+1}, \underline{\mathbf{w}}_{k+1}^h\right) = \underline{\mathbf{0}}, \tag{8}$$

where $\underline{\mathbf{w}}^h = \mathrm{vec}(\underline{\mathbf{u}}^h, \underline{\mathbf{p}}^h) \in \mathbb{R}^{N_u^h + N_p^h}$ and

$$H^h := \begin{bmatrix} M^h & \\ & \end{bmatrix}, \quad \mathring{\underline{\mathbf{f}}}^h\left(t, \underline{\mathbf{w}}^h\right) := \begin{bmatrix} \underline{\mathbf{f}}^h(t) \\ \underline{\mathbf{0}} \end{bmatrix} - \begin{bmatrix} K^h + C^h\left(\underline{\mathbf{u}}^h\right) & (D^h)^\mathrm{T} \\ D^h & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^h \\ \underline{\mathbf{p}}^h \end{bmatrix}.$$

The coefficients $\{\alpha_j\}_{j=1}^{\sigma}$ and β depend on the specific BDF scheme. For example, for $\sigma = 1$ we have $\alpha_1 = \beta = 1$ (Backward Euler scheme), and, for $\sigma = 2$, $\alpha_1 = 4/3$, $\alpha_2 = -1/3$ and $\beta = 2/3$; these two choices lead to numerical methods of first and second order, respectively.

Eq. (8) is in general nonlinear and its solution requires the application of ad-hoc numerical methods; in this paper, we adopt the Newton–Raphson algorithm (see Section 4.3).

4. Modular domain-decomposition of arteries

With the aim of the model order reduction presented in Section 5, it is beneficial to perform a geometrical approximation of the vessel based on a domain-decomposition approach. We first set the theoretical basis for the approximation of the Navier–Stokes equations on modular geometries in Section 4.1. The discretization of the continuous formulation is then performed in Section 4.2, where we also present our approach for the treatment of the coupling variables (i.e., Lagrange multipliers) at the interfaces.

4.1. The continuous Navier–Stokes equations on modular geometries

We introduce a library of *building blocks* $\widehat{\Omega}^i$, $i = 1, \dots, N_\mathrm{bb}$. In the context of cardiovascular simulations, these reference building blocks are model cylinders and bifurcations, as shown in Fig. 1. The target geometry is then approximated as a modular composition of subdomains $\Omega \approx \Omega_\mathrm{m}(\mathscr{M}) = \bigcup_{j=1}^{N_\Omega} \Omega^j(\underline{\boldsymbol{\mu}}^j)$. Here, $\Omega^j := \Phi^{z(j)}(\widehat{\Omega}^{z(j)}; \underline{\boldsymbol{\mu}}^j)$ is an open and bounded subdomain obtained by applying a prescribed parametrized geometrical deformation $\Phi^{z(j)}$ to the $z(j)$th building block, $z : [1, \dots, N_\Omega] \mapsto [1, \dots, N_\mathrm{bb}]$ is a map from the indices of the subdomains in the target geometry to the indices of the building blocks, and $\mathscr{M} := \{\underline{\boldsymbol{\mu}}^j\}_{j=1}^{N_\Omega}$ is the set of geometrical parameters. We remark that, even though the RB method is suitable to handle geometrical and physical parameters, in the remainder of the paper the symbol $\underline{\boldsymbol{\mu}}$ refers exclusively to geometrical ones. The case in which, for instance, the viscosity and density of blood are parameters of the problem will be addressed in future extensions of the present work. In the following, we indicate $z(j) := z_j$ for brevity. Each vector of parameters $\underline{\boldsymbol{\mu}}^j$ belongs to a space $\mathcal{D}^{z_j} \subset \mathbb{R}^{N_\mu^{z_j}}$

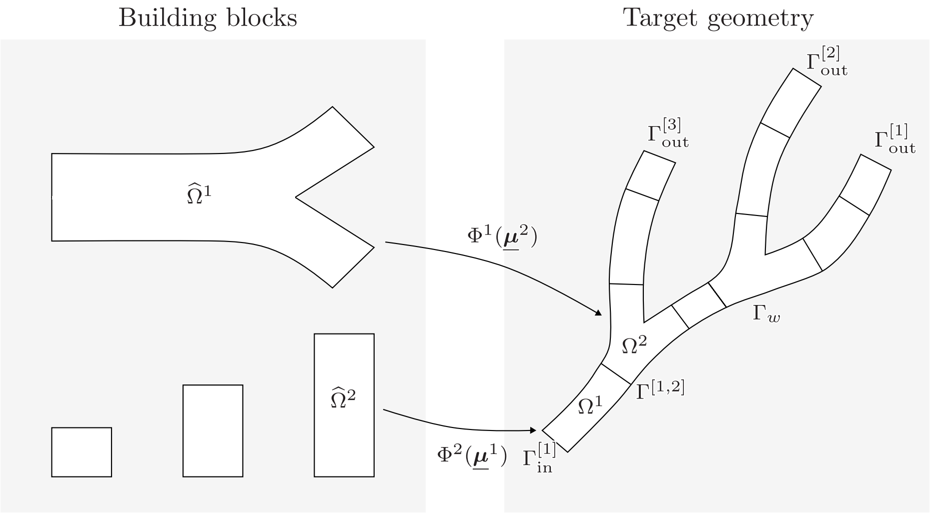
Building blocks
Target geometry
$\widehat{\Omega}^1$
$\widehat{\Omega}^2$
$\Phi^1(\underline{\boldsymbol{\mu}}^2)$
$\Phi^2(\underline{\boldsymbol{\mu}}^1)$
$\Gamma_{\mathrm{out}}^{[2]}$
$\Gamma_{\mathrm{out}}^{[3]}$
$\Gamma_{\mathrm{out}}^{[1]}$
Γ_w
Ω^2
Ω^1
$\Gamma^{[1,2]}$
$\Gamma_{\mathrm{in}}^{[1]}$

Fig. 1. Sketch of the domain-decomposition of a target geometry. Each block in the target geometry is found from the parametrized geometrical deformation of a small number of reference building blocks.

whose dimensionality depends on the corresponding reference building block. For each $i = 1, \ldots, N_{\mathrm{bb}}$ and given a parameter vector $\underline{\widetilde{\boldsymbol{\mu}}} \in \mathbb{R}^{N_\mu^i}$, we focus on geometrical deformations of the form

$$\Phi^i(\widehat{\mathbf{x}}; \underline{\widetilde{\boldsymbol{\mu}}}) = Q(\underline{\boldsymbol{\mu}})\boldsymbol{\chi}^i(\widehat{\mathbf{x}}; \underline{\widetilde{\boldsymbol{\mu}}}) + \mathbf{t}(\underline{\widetilde{\boldsymbol{\mu}}}), \quad \forall \widehat{\mathbf{x}} \in \widehat{\Omega}^i, \tag{9}$$

where $Q(\underline{\widetilde{\boldsymbol{\mu}}})$ is a rotation matrix, $\mathbf{t}$ is a translation vector, and $\boldsymbol{\chi}^i(\cdot; \underline{\widetilde{\boldsymbol{\mu}}})$ is a nonaffine geometrical deformation. Specifically, the nonaffine deformation $\boldsymbol{\chi}^i(\cdot; \underline{\widetilde{\boldsymbol{\mu}}})$ is particular to the type of building block corresponding to index i. The types of building blocks we consider and the corresponding admissible nonaffine deformations are depicted in Fig. 2: for the tubes, we consider functions $\boldsymbol{\chi}^i(\cdot; \underline{\widetilde{\boldsymbol{\mu}}})$ able to vary their length, outlet radius, and bending angle, and for the bifurcations we are interested in rotating the normals of the outlets.

The subdomains in the decomposed geometry do not overlap, i.e., $\Omega^l(\underline{\boldsymbol{\mu}}^l) \cap \Omega^m(\underline{\boldsymbol{\mu}}^m) = \emptyset$ if $l \neq m$. We define the interface between two subdomains $\Gamma^{[jm]}(\underline{\boldsymbol{\mu}}^j, \underline{\boldsymbol{\mu}}^m) := \overline{\Omega}^j(\underline{\boldsymbol{\mu}}^j) \cap \overline{\Omega}^m(\underline{\boldsymbol{\mu}}^m)$. The building blocks in the reference configuration are designed with circular inlet and outlet faces; the geometrical deformations are chosen such that the interfaces are circles for every possible choice of the geometrical parameters.

Remark 3. Although $\Gamma^{[lm]}$ and $\Gamma^{[ml]}$ represent the same physical surface, it is still beneficial to differentiate between the two as we associate with each interface the vectors $\mathbf{n}_{lm}$ and $\mathbf{n}_{ml}$, i.e., the outward normal unit vectors with respect to Ω^l and Ω^m (clearly, $\mathbf{n}_{lm} = -\mathbf{n}_{ml}$). This distinction allows to simplify the notation in W2.

For every subdomain $\Omega^j(\underline{\boldsymbol{\mu}}^j)$, we introduce the set of indices of the neighboring subdomains $N(j)$ and the sets $I_{\mathrm{in}}(j)$ and $I_{\mathrm{out}}(j)$, which contain the indices of the inlet and outlet interfaces whose intersection with the boundary of $\Omega^j(\underline{\boldsymbol{\mu}}^j)$ is nonempty. More precisely, $\partial\Omega^j(\underline{\boldsymbol{\mu}}^j) \cap \Gamma_{\mathrm{in}}^{[i]} \neq \emptyset$ for all $i \in I_{\mathrm{in}}(j)$, and $\partial\Omega^j(\underline{\boldsymbol{\mu}}^j) \cap \Gamma_{\mathrm{out}}^{[i]} \neq \emptyset$ for all $i \in I_{\mathrm{out}}(j)$. The dependance of the subdomains Ω^j and interfaces $\Gamma^{[jm]}$, $\Gamma_{\mathrm{in}}^{[i]}$ and $\Gamma_{\mathrm{out}}^{[i]}$ on the geometrical parameters $\mathscr{M}$ is omitted unless ambiguity arises in the remainder of the paper.

Let us rewrite the weak formulation W1 to account for the modular decomposition of the original domain, i.e., Ω_{m}. For each subdomain Ω^j, we introduce the spaces $\mathcal{V}_g^j := [H_{g,\Gamma_D}^1(\Omega^j)]^d$, $\mathcal{Q}^j := L^2(\Omega^j)$ and $\mathcal{V}_0^j := [H_{0,\Gamma_D}^1(\Omega^j)]^d$. Moreover, for every interface $\Gamma^{[jm]}$ we define the spaces $\mathcal{L}^{[jm]} = [H_{00}^{-1/2}(\Gamma^{[jm]})]^d$. For the sake of conciseness, let us use the following notation

$$\begin{aligned}
\mathscr{M}^j(\boldsymbol{\varepsilon}, \boldsymbol{\varphi}, \psi; \boldsymbol{\omega}) &:= \int_{\Omega^j} \rho_{\mathrm{f}}\boldsymbol{\varepsilon} \cdot \boldsymbol{\omega} + \int_{\Omega^j} \rho_{\mathrm{f}}[(\boldsymbol{\varphi} \cdot \nabla)\boldsymbol{\varphi}] \cdot \boldsymbol{\omega} + \int_{\Omega^j} \sigma(\boldsymbol{\varphi}, \psi) : \nabla\boldsymbol{\omega} \\
\mathscr{C}^j(\boldsymbol{\varphi}; \eta) &:= \int_{\Omega^j} \nabla \cdot \boldsymbol{\varphi}\eta,
\end{aligned}$$

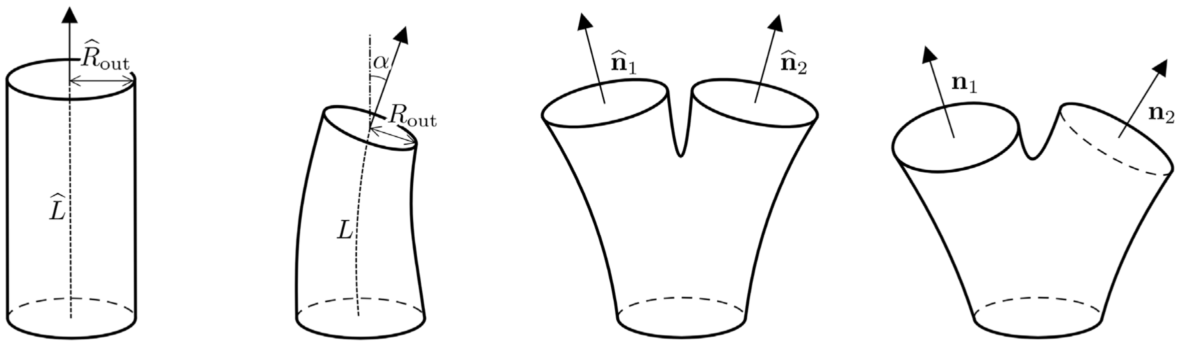

Fig. 2. Types of reference building blocks and affine transformations. On the left, tubes: the geometrical parameters are the angle of the outlet normal α—due to the axial symmetry and to the rotation matrix Q in Eq. (9), a single angle is sufficient to represent a bending in any direction—in the deformed configuration α and ratios between the reference and deformed lengths ($\widehat{L}/L$) and reference radii ($\widehat{R}/R$). On the right, bifurcation: the geometrical parameters are the angles describing the rotation of the reference outlet normals $\widehat{\mathbf{n}}_1$ and $\widehat{\mathbf{n}}_2$ onto the outlet normals $\mathbf{n}_1$ and $\mathbf{n}_2$ (i.e., three Euler angles per outlet, that is six geometrical parameters in total).

for every $\boldsymbol{\varepsilon}, \boldsymbol{\varphi}, \boldsymbol{\omega} \in [H^1(\Omega^j)]^d$ and for every $\psi, \eta \in L^2(\Omega^j)$. Assuming for simplicity that $\Omega = \Omega_{\mathrm{m}}$, it can be shown that W1 is equivalent—in a sense that will specified in Remark 4—to the following weak formulation:

(**W2**) *given* $\mathbf{f}$, $\mathbf{g}$, *and* $\mathbf{h}$ *regular enough and for every* $j = 1, \dots, N_\Omega$, *find* $(\mathbf{u}^j, p^j, \{\boldsymbol{\lambda}^{[jm]}\}_{m \in N(j)}) \in \mathcal{V}_g^j \times \mathcal{Q}^j \times \prod_{m \in N(j)} \mathcal{L}^{[jm]}$, *such that, for every* $t \in (0, T)$,

$$\begin{aligned} \mathscr{M}^j\left(\frac{\partial \mathbf{u}^j}{\partial t}, \mathbf{u}^j, p^j; \mathbf{v}\right) + \sum_{m \in N(j)} \int_{\Gamma^{[jm]}} \boldsymbol{\lambda}^{[jm]} \cdot \mathbf{v} = \int_{\Omega^j} \mathbf{f} \cdot \mathbf{v} + \sum_{i \in I_{\text{out}}(j)} \int_{\Gamma_{\text{out}}^{[i]}} \mathbf{h} \cdot \mathbf{v} \qquad & \forall \mathbf{v} \in \mathcal{V}_0^j, \\ \mathscr{C}^j\left(\mathbf{u}^j; q\right) = 0 \qquad & \forall q \in \mathcal{Q}^j, \end{aligned} \tag{10}$$

and such that $\mathbf{u}^j = \mathbf{u}_0|_{\Omega^j}$ *for* $t = 0$. *Moreover, for every* $m \in N(j)$, $\boldsymbol{\lambda}^{[jm]} = -\boldsymbol{\lambda}^{[mj]}$ *and*

$$\int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot \left(\mathbf{u}^j - \mathbf{u}^m\right) = 0 \qquad \forall \boldsymbol{\eta} \in \mathcal{L}^{[jm]}. \tag{11}$$

Remark 4. The two weak formulations W1 and W2 are equivalent in the following sense. If $(\mathbf{u}, p)$ satisfies W1, then $(\mathbf{u}|_{\Omega^j}, p|_{\Omega^j}, \{\sigma(\mathbf{u}, p)\mathbf{n}_{jm}\}_{m \in N(j)})$ is also solution of the local problem W2, for every $j = 1, \dots, N_\Omega$. The Lagrange multipliers therefore play the role of the stress at the interfaces; for details, see, e.g., [25,43]. Conversely, if $(\mathbf{u}^j, p^j, \{\boldsymbol{\lambda}^{[jm]}\}_{m \in N(j)})$ are the local solutions of W2, then $(\mathbf{u}, p) = (\Pi_{j=1}^{N_\Omega} \mathbf{u}^j, \Pi_{j=1}^{N_\Omega} p^j)$ is solution of W1.

4.2. Discretization of the primal hybrid formulation of the flow problem

The discretization of differential problems by the FE method requires the definition of a computational mesh, as already mentioned in Section 3.2. In the case of the approximated modular geometry Ω_{m}, each building block $\widehat{\Omega}^i$ is equipped with its own triangulation $\widehat{\mathscr{T}}^{i,h}$. Therefore, the global mesh $\mathscr{T}_{\mathrm{m}}^h(\mathscr{M}) = \bigcup_{j=1}^{\mathrm{N}_\Omega} \mathscr{T}^{j,h}(\underline{\boldsymbol{\mu}}^j) = \bigcup_{j=1}^{\mathrm{N}_\Omega} \Phi^{z_j}(\widehat{\mathscr{T}}^{z_j,h}; \underline{\boldsymbol{\mu}}^j)$ is a composition of distinct meshes which do not necessarily satisfy conformity constraints at the interfaces. We recall that a mesh is conforming if, for every two elements, their intersection is either empty or a face, edge or vertex. In other words, a conforming mesh does not feature any *hanging nodes*.

The global mesh is in general nonconforming and we are compelled to consider nonconforming domain-decomposition methods for the solution of the Navier–Stokes equations. These are formally defined as domain-decomposition methods in which the search space for the discrete solution is not a subset of the continuous search space (in our case $\mathcal{V}_g \times \mathcal{Q}$). The most popular approaches rely on the introduction of suitable Lagrange multipliers at the interfaces to enforce transmission conditions, as in W2; see, e.g., the well-known mortar method [44–46] and INTERNODES [47,48]. In this paper, we adopt the algorithm presented in [25], which is based on the discretization of the Lagrange multipliers space via a small number of spectral basis functions defined on the interfaces. For our application, this choice is convenient because (i) the method allows us to recover the h-convergence order of

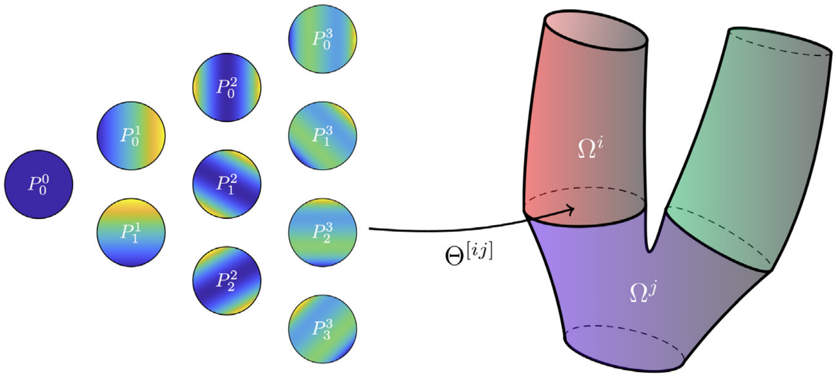

Fig. 3. Basis functions P_k^n on the unit disk $\mathscr{D}$, for $n \leq 3$, mapped onto the target interface $\Gamma^{[ij]}$. Minimum and maximum values are plotted in blue and yellow, respectively. The basis function P_0^0 is constant. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the primal discretization method—i.e., the FE method—even when a small number of spectral basis functions is considered, (ii) defining a spectral basis on each interface does not require to project nor to interpolate the traces of FE basis functions from one side to the other, and (iii) as already mentioned, the interfaces are circular in the target configuration, which allows us to employ a set of standard orthonormal basis functions on the two-dimensional disk. In the remainder of this section and in Section 4.2 we recall the basics of this nonconforming method applied to the Navier–Stokes equations; the interested reader is referred to [25] for the details.

We follow the same procedure presented in Section 3.2 for the discretization of the local variables $\mathbf{u}^j$ and p^j, which become $\mathbf{u}^{j,h} = \sum_{i=1}^{N_u^{z_j,h}} u_i^{j,h} \boldsymbol{\varphi}_i^{j,h}$ and $p^{j,h} = \sum_{i=1}^{N_p^{z_j,h}} p_i^{j,h} \psi_i^{j,h}$. We remark that—since the meshes $\mathscr{T}^{j,h}$ are generated as transformations of the reference ones $\widehat{\mathscr{T}}^{z_j,h}$—the number of nodes $N_u^{z_j,h}$ and $N_p^{z_j,h}$ are indexed with respect to the reference building blocks; moreover, we have that $\boldsymbol{\varphi}_i^{j,h} := \widehat{\boldsymbol{\varphi}}^{z_j,h} \circ (\Phi^{z_j}(\underline{\boldsymbol{\mu}}^j))^{-1}$, i.e., the velocity FE basis functions on the subdomains are obtained as a composition of the reference velocity FE basis functions $\widehat{\boldsymbol{\varphi}}^{z_j,h}$ with the inverse of the map $\Phi^{z_j}(\underline{\boldsymbol{\mu}}^j)$ (the same holds true for the pressure FE basis functions $\psi_i^{j,h}$). For the Lagrange multiplier $\boldsymbol{\lambda}^{[jm]}$ we consider the approximation $\boldsymbol{\lambda}^{[jm],\delta} = \sum_{i=1}^{N_\lambda} \lambda_i^{[jm],\delta} \boldsymbol{\xi}_i^{[jm],\delta}$, where $\boldsymbol{\xi}_i^{[jm],\delta} = c_{jm} \widehat{\boldsymbol{\xi}}_i^{\delta} \circ (\Theta^{[jm]})^{-1} \in [L^2(\Gamma^{[jm]})]^d$, $c_{jm} \in \mathbb{R}$, $\widehat{\boldsymbol{\xi}}_i^{\delta} \in [L^2(\mathscr{D})]^d$ are a set of orthogonal functions on the unit disk $\mathscr{D} \in \mathbb{R}^{d-1}$ and $\Theta^{[jm]} : \mathscr{D} \mapsto \Gamma^{[jm]}$ is a bijective map from the unit disk $\mathscr{D}$ to the target interface $\Gamma^{[jm]}$. Similarly, we define the maps $\Theta_{\text{in}}^{[m]} : \mathscr{D} \mapsto \Gamma_{\text{in}}^{[m]}$ from the unit disk $\mathscr{D}$ to the inlet interfaces $\Gamma_{\text{in}}^{[m]}$, the basis functions $\boldsymbol{\xi}_{i,\text{in}}^{[m],\delta} = \widehat{\boldsymbol{\xi}}_i^{\delta} \circ (\Theta_{\text{in}}^{[m]})^{-1} \in [L^2(\Gamma_{\text{in}}^{[m]})]^d$, and $\boldsymbol{\lambda}_{\text{in}}^{[m],\delta} = \sum_{i=1}^{N_\lambda} \lambda_{i,\text{in}}^{[m],\delta} \boldsymbol{\xi}_{i,\text{in}}^{[m],\delta}$. These functions are used to set the inlet velocity profiles, allowing one to effortlessly transition from the FE model to the reduced one through the process described in Section 5. The use of Lagrange multipliers is a classical way to weakly impose Dirichlet boundary conditions (see, e.g., [49]) and, compared to other popular approaches such as penalty methods [50,51], it has the advantage of being variationally consistent. We remark that (i) we introduce the discretization parameter δ for the Lagrange multipliers to indicate that the degree of refinement is in fact independent of the mesh size in $\mathscr{T}^{j,h}$ or $\mathscr{T}^{m,h}$, and (ii) we consider for simplicity the same number N_λ of basis functions at each interface.

In this paper, we build the basis functions at the interfaces $\{\widehat{\boldsymbol{\xi}}_i^{\delta}\}_{i=1}^{N_\lambda}$ using the scalar functions P_k^n shown in Fig. 3. These are constructed as follows. Let us consider Chebyshev polynomials of the second kind U_n, which are defined through the recurrence relation $U_0(x) = 1$, $U_1(x) = 2x$, $U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$. Then, for $0 \leq k \leq n$,

$$P_k^n(x, y) = \frac{1}{\sqrt{\pi}} U_n \left(x \cos(\omega x) + y \sin(\omega y) \right), \quad \omega = \frac{k}{n+1}\pi,$$

are orthonormal polynomials on the unit disk $\mathscr{D}$ with respect to the weight function $W(x, y) = 1/\sqrt{\pi}$ [52]. Given $n \geq 0$, we set

$$\widehat{\Xi}_n := \{\widehat{\boldsymbol{\xi}^{\delta}}_i\}_{i=1}^{N_\lambda} = \bigcup_{i=1}^{d} \bigcup_{\tilde{n}=0}^{n} \bigcup_{k=0}^{\tilde{n}} P_k^{\tilde{n}} \mathbf{e}_i,$$

where $\mathbf{e}_i$ is the ith canonic vector. It is trivial to find that $N_\lambda = d(n+1)(n+2)/2$.

Let us now address the discretization of the individual elements of W2, which is local to subdomain Ω^j, the assembly of the global block system, and the discretization in time.

Discretization of the coupled momentum and continuity equations. For every $j = 1, \ldots, N_\Omega$ and $m \in N(j)$, let us define $\underline{\lambda}^{[jm],\delta} = [\lambda_1^{[jm],\delta}, \ldots, \lambda_{N_\lambda}^{[jm],\delta}]$ and the coupling matrix $B_{pq}^{[jm],h\delta} = \int_{\Gamma^{[jm]}} \boldsymbol{\xi}_p^{[jm],\delta} \cdot \boldsymbol{\varphi}_q^{j,h}$. Then, Eq. (10) can be rewritten in algebraic form as

$$\begin{bmatrix} M^{j,h} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \dot{\underline{\mathbf{u}}}^{j,h} \\ \dot{\underline{\mathbf{p}}}^{j,h} \\ \dot{\underline{\boldsymbol{\lambda}}}^{j,\delta} \end{bmatrix} + \begin{bmatrix} K^{j,h} + C^{j,h}\left(\underline{\mathbf{u}}^{j,h}\right) & (D^{j,h})^{\mathrm{T}} & (B^{j,h\delta})^{\mathrm{T}} \\ D^{j,h} & & \\ & & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{j,h} \\ \underline{\mathbf{p}}^{j,h} \\ \underline{\boldsymbol{\lambda}}^{j,\delta} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}}^{j,h} \\ \underline{\mathbf{0}} \end{bmatrix}. \tag{12}$$

In Eq. (12), we denoted by $\underline{\boldsymbol{\lambda}}^{j,\delta}$ the vector containing all $\underline{\boldsymbol{\lambda}}^{[jm],\delta}$ and by $B^{j,h\delta}$ the matrix obtained by stacking the various $B^{[jm],h\delta}$, for every $m \in N(j)$.

Equality of the Lagrange multipliers at the same interface. A natural way to enforce the constraint $\boldsymbol{\lambda}^{[jm]} = -\boldsymbol{\lambda}^{[mj]}$ in W2 is to choose $c_{jm} = 1 = -c_{mj}$ such that $\boldsymbol{\xi}_i^{[jm],\delta} = -\boldsymbol{\xi}_i^{[mj],\delta}$ and $\lambda_i^{[jm],\delta} = \lambda_i^{[mj],\delta}$ for all $m \in N(j)$ and for $i = 1, \ldots, N_\lambda$. As a consequence, it is legitimate to introduce a numbering of the interfaces $\Gamma^{[1]}, \ldots, \Gamma^{[N_\Gamma]}$ and to denote the corresponding (unique) vectors of degrees of freedom of the Lagrange multipliers by $\underline{\boldsymbol{\lambda}}^{[1],\delta}, \ldots,$ $\underline{\boldsymbol{\lambda}}^{[N_\Gamma],\delta}$. Furthermore, it holds that $B_{pq}^{[mj],h\delta} = -\int_{\Gamma^{[mj]}} \boldsymbol{\xi}_p^{[jm],\delta} \cdot \boldsymbol{\varphi}_q^{m,h}$. In the following, the coupling matrix between the ith interface and the velocity in the jth subdomain is also written $B^{[i]j,h\delta}$ (to be intended as null matrix, if $\Gamma^{[i]} \cap \overline{\Omega}^j = \emptyset$).

Discretization of the weak continuity equation. The numerical discretization of Eq. (11) entails the definition of a preferential side for $\Gamma^{[jm]}$ which determines the computational mesh and the FE basis functions to be used in the evaluation of the integral. However, the direct approximation of Eq. (11) is problematic because it requires the projection or interpolation of the velocity from one side of the interface to the other (such operation is required in methods such as the mortar method or INTERNODES, for instance). At the continuous level, it is evidently possible to write

$$\int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot \left(\mathbf{u}^j - \mathbf{u}^m\right) = \int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot \mathbf{u}^j - \int_{\Gamma^{[mj]}} \boldsymbol{\eta} \cdot \mathbf{u}^m = 0. \tag{13}$$

The advantage of separating the integral into the two contributions on $\Gamma^{[jm]}$ and $\Gamma^{[mj]}$ is apparent at the discrete level. Indeed, by substituting $\boldsymbol{\eta}$ with $\boldsymbol{\xi}_i^{[jm],\delta}$ and $-\boldsymbol{\xi}_i^{[jm],\delta}$ in the two integrals on the right hand side of Eq. (13), Eq. (11) is discretized as

$$B^{[jm],h\delta}\underline{\mathbf{u}}^j + B^{[mj],h\delta}\underline{\mathbf{u}}^m = \underline{\mathbf{0}}.$$

Weak imposition of Dirichlet boundary conditions. We recall that we assume that every inlet interface $\Gamma_{\text{in}}^{[m]}$ correspond to a single subdomain Ω^j. The condition $\mathbf{u}^{j,h} = \mathbf{g}_m$ on $\Gamma_{\text{in}}^{[m]}$ is weakly imposed as

$$\int_{\Gamma_{\text{in}}^{[m]}} \boldsymbol{\xi}_{i,\text{in}}^{[m],\delta} \cdot \left(\mathbf{u}^{j,h} - \mathbf{g}_m\right) = 0, \quad i = 1, \ldots, N_\lambda. \tag{14}$$

By introducing the interpolation of the boundary data $\mathbf{g}_m$ onto the FE space spanned by $\{\boldsymbol{\varphi}_i^{j,h}\}_{i=1}^{N_u^{z\,j,h}}$, the corresponding vector of degrees of freedom $\underline{\mathbf{g}}_m^h$, and $(B_{\text{in}}^{[m]j,h\delta})_{pq} = \int_{\Gamma_{\text{in}}^{[m]}} \boldsymbol{\xi}_{p,\text{in}}^{[m],\delta} \cdot \boldsymbol{\varphi}_q^{j,h}$, Eq. (14) is approximated as

$$B_{\text{in}}^{[m]j,h\delta}\underline{\mathbf{u}}^{j,h} - B_{\text{in}}^{[m]j,h\delta}\underline{\mathbf{g}}_m^h = \underline{\mathbf{0}},$$

and the first equation in Eq. (12) is accordingly modified as

$$M^{j,h}\dot{\underline{\mathbf{u}}}^{j,h} + \left(K^{j,h} + C^{j,h}(\underline{\mathbf{u}}^{j,h})\right)\underline{\mathbf{u}}^{j,h} + (D^{j,h})^{\mathrm{T}}\underline{\mathbf{p}}^{j,h} + (B^{j,h\delta})^{\mathrm{T}}\underline{\boldsymbol{\lambda}}^{j,h} + (B_{\text{in}}^{[m]j,h\delta})^{\mathrm{T}}\boldsymbol{\lambda}_{\text{in}}^{[m],\delta} = \underline{\mathbf{f}}$$

to account for the Lagrange multiplier $\boldsymbol{\lambda}_{\text{in}}^{[m],\delta}$.

Assembly of global system. It is possible to arrange the local systems corresponding to the subdomains in the form of a global block system as

$$\begin{bmatrix} \mathcal{M}^h & \\ & \end{bmatrix} \begin{bmatrix} \dot{\underline{\mathbf{W}}}^h \\ \dot{\underline{\boldsymbol{\Lambda}}}^\delta \end{bmatrix} + \begin{bmatrix} \mathcal{A}^h(\underline{\mathbf{W}}^h) & (\mathcal{B}^{h\delta})^{\mathrm{T}} \\ \mathcal{B}^{h\delta} & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{W}}^h \\ \underline{\boldsymbol{\Lambda}}^\delta \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{F}}^h \\ \underline{\mathbf{G}}^{h\delta} \end{bmatrix},$$

where

$$\mathcal{M}^h := \operatorname{diag}\left(\begin{bmatrix} M^{j,h} & \\ & \end{bmatrix}\right)_{j=1,\dots,N_\Omega}, \qquad \mathcal{A}^h(\underline{\mathbf{W}}^h) := \operatorname{diag}\left(\begin{bmatrix} K^{j,h}+C^{j,h}(\underline{\mathbf{u}}^{j,h}) & (D^{j,h})^{\mathrm{T}} \\ D^{j,h} & \end{bmatrix}\right)_{j=1,\dots,N_\Omega},$$

$\mathcal{B}^{h\delta}$ is a block matrix such that $(\mathcal{B}^{h\delta})_{ij} = [B^{[i]j,h\delta}, O]$ (O being the null matrix) if $i \leq N_\Gamma$ and $(\mathcal{B}^{h\delta})_{ij} = [B_{\mathrm{in}}^{[i-N_\Gamma]j,h\delta}, O]$ otherwise,

$$\underline{\mathbf{W}}^h = \operatorname{vec}(\underline{\mathbf{w}}^{1,h}, \dots, \underline{\mathbf{w}}^{N_\Omega,h}), \quad \underline{\boldsymbol{\Lambda}}^\delta = \operatorname{vec}(\underline{\boldsymbol{\lambda}}^{[1],\delta}, \dots, \underline{\boldsymbol{\lambda}}^{[N_\Gamma],\delta}, \underline{\boldsymbol{\lambda}}_{\mathrm{in}}^{[1],\delta}, \dots, \underline{\boldsymbol{\lambda}}_{\mathrm{in}}^{[N_{\mathrm{in}}],\delta}),$$

and $\underline{\mathbf{F}}^h$ and $\underline{\mathbf{G}}^{h\delta}$ are block vectors accounting for the forcing terms and Dirichlet boundary conditions, respectively. Operator "diag" returns a diagonal matrix where each argument with index j is placed in the jth diagonal block.

Discretization in time and global nonlinear residual. The discretization in time is performed along the same lines of the discussion in Section 3.2. We define $\underline{\mathbf{Y}}^{h\delta} = \operatorname{vec}(\underline{\mathbf{W}}^h, \underline{\boldsymbol{\Lambda}}^\delta)$ and

$$\mathcal{H}^h := \begin{bmatrix} \mathcal{M}^h & \\ & \end{bmatrix}, \quad \overset{\circ}{\underline{\mathbf{F}}}{}^{h\delta}(t, \underline{\mathbf{Y}}) := \begin{bmatrix} \underline{\mathbf{F}}^h(t) \\ \underline{\mathbf{G}}^{h\delta}(t) \end{bmatrix} - \begin{bmatrix} \mathcal{A}^h(\underline{\mathbf{W}}^h) & (\mathcal{B}^{h\delta})^{\mathrm{T}} \\ \mathcal{B}^{h\delta} & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{W}}^h \\ \underline{\boldsymbol{\Lambda}}^\delta \end{bmatrix}.$$

Then, given $\underline{\mathbf{Y}}_{k-j+1}^{h\delta}$ for $j = 1, \dots, \sigma$, the solution at timestep t_{k+1} is found by solving

$$\underline{\mathbf{R}}\left(\underline{\mathbf{Y}}_{k+1}^{h\delta}\right) := \mathcal{H}^h \underline{\mathbf{Y}}_{k+1}^{h\delta} - \sum_{j=1}^{\sigma} \alpha_j \mathcal{H}^h \underline{\mathbf{Y}}_{k-j+1}^{h\delta} - \Delta t \beta \overset{\circ}{\underline{\mathbf{F}}}{}^{h\delta}\left(t_{k+1}, \underline{\mathbf{Y}}_{k+1}^{h\delta}\right) = \underline{\mathbf{0}}. \tag{15}$$

4.3. Efficient solution of the global nonlinear system

Eq. (15) is nonlinear and hence solved using the Newton–Raphson algorithm. In particular, given an initial guess $\underline{\mathbf{Y}}^{(0)}$, the $(l+1)$th iteration of the algorithm for the solution of $\underline{\mathbf{R}}(\underline{\mathbf{Y}}) = \underline{\mathbf{0}}$ is

$$\underline{\mathbf{Y}}^{(l+1)} = \underline{\mathbf{Y}}^{(l)} - \left(\mathcal{J}_{\underline{\mathbf{R}}}(\underline{\mathbf{Y}}^{(l)})\right)^{-1} \underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(l)}), \tag{16}$$

where $\mathcal{J}_{\underline{\mathbf{R}}}$ is the tangent matrix of $\underline{\mathbf{R}}$. The stopping criterion is based on a user-provided tolerance τ_{NR} and reads $\|\underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(l)})\|_2 / \|\underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(0)})\|_2 < \tau_{\mathrm{NR}}$.

In order to efficiently solve the linear system in Eq. (16) via iterative methods such as GMRES [53], we need to develop a preconditioner for the tangent matrix $\mathcal{J}_{\underline{\mathbf{R}}}$. Differentiating Eq. (15) with respect to its only argument yields

$$\mathcal{J}_{\underline{\mathbf{R}}}(\underline{\mathbf{Y}}) = \begin{bmatrix} \mathcal{M} + \Delta t \beta \mathcal{A}^h(\underline{\mathbf{W}}) & \Delta t \beta (\mathcal{B}^{h\delta})^{\mathrm{T}} \\ \Delta t \beta \mathcal{B}^{h\delta} & \end{bmatrix} = \begin{bmatrix} \widetilde{\mathcal{A}}(\underline{\mathbf{W}}) & \widetilde{\mathcal{B}}^{\mathrm{T}} \\ \widetilde{\mathcal{B}} & \end{bmatrix}. \tag{17}$$

Thus, the tangent matrix features a saddle-point structure stemming directly from the original differential problem W2—which is in fact a saddle-point problem. In the remainder of this section, we omit the explicit dependence of $\mathcal{J}_{\underline{\mathbf{R}}}$ and $\widetilde{\mathcal{A}}$ on $\underline{\mathbf{Y}}$ and $\underline{\mathbf{W}}$, respectively, for the sake of clarity of notation. A possible strategy to design a preconditioner is based on the (exact) decomposition

$$\begin{bmatrix} \widetilde{\mathcal{A}} & \widetilde{\mathcal{B}}^{\mathrm{T}} \\ \widetilde{\mathcal{B}} & \end{bmatrix} = \begin{bmatrix} \mathcal{I} & \\ \widetilde{\mathcal{B}}\widetilde{\mathcal{A}}^{-1} & \mathcal{I} \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{A}} & \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \widetilde{\mathcal{A}}^{-1}\widetilde{\mathcal{B}}^{\mathrm{T}} \\ & \mathcal{I} \end{bmatrix}, \tag{18}$$

where $\mathcal{S} = -\widetilde{\mathcal{B}}\widetilde{\mathcal{A}}^{-1}\widetilde{\mathcal{B}}^{\mathrm{T}}$ is the Schur complement of $\mathcal{J}_{\underline{\mathbf{R}}}$. This decomposition is the foundation of several preconditioners for saddle-point systems, such as SIMPLE [54] and the nested block preconditioner for blood flow simulations proposed in [55]. The solution of a linear system of the form $\mathcal{J}_{\underline{\mathbf{R}}}\underline{\mathbf{X}} = \underline{\mathbf{B}}$, with $\underline{\mathbf{X}} = \operatorname{vec}(\underline{\mathbf{X}}_w, \underline{\mathbf{X}}_\lambda)$ and $\underline{\mathbf{B}} = \operatorname{vec}(\underline{\mathbf{B}}_w, \underline{\mathbf{B}}_\lambda)$, amounts to solving

$$\begin{bmatrix} \widetilde{\mathcal{A}} & \widetilde{\mathcal{B}}^{\mathrm{T}} \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{X}}_w \\ \underline{\mathbf{X}}_\lambda \end{bmatrix} = \begin{bmatrix} \widetilde{\mathcal{A}} & \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \widetilde{\mathcal{A}}^{-1}\widetilde{\mathcal{B}}^{\mathrm{T}} \\ & \mathcal{I} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{X}}_w \\ \underline{\mathbf{X}}_\lambda \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{B}}_w \\ \underline{\mathbf{B}}_\lambda - \widetilde{\mathcal{B}}\widetilde{\mathcal{A}}^{-1}\underline{\mathbf{B}}_w \end{bmatrix}, \tag{19}$$

hence $\underline{\mathbf{X}}_\lambda = \mathcal{S}^{-1}(\underline{\mathbf{B}}_\lambda - \widetilde{\mathcal{B}}\underline{\mathbf{Z}}_w)$ and $\underline{\mathbf{X}}_w = \underline{\mathbf{Z}}_w - \widetilde{\mathcal{A}}^{-1}\widetilde{\mathcal{B}}^{\mathrm{T}}\underline{\mathbf{X}}_\lambda$, where $\underline{\mathbf{Z}}_w = \widetilde{\mathcal{A}}^{-1}\underline{\mathbf{B}}_w$ can be computed only once for efficiency.

The bottlenecks of the algorithm presented above are evidently the computation of the Schur complement $\mathcal{S}$ and the inversion of matrices $\widetilde{\mathcal{A}}$ and $\mathcal{S}$.

Block (i, j) of the Schur complement explicitly takes the form

$$(\mathcal{S})_{ij} = -\sum_{l=1}^{N_\Omega}\sum_{m=1}^{N_\Omega}(\widetilde{\mathcal{B}})_{il}(\widetilde{\mathcal{A}}^{-1})_{lm}(\widetilde{\mathcal{B}}^{\mathrm{T}})_{mj} = -\sum_{k=1}^{N_\Omega}(\widetilde{\mathcal{B}})_{ik}(\widetilde{\mathcal{A}}^{-1})_{kk}(\widetilde{\mathcal{B}}^{\mathrm{T}})_{kj}, \tag{20}$$

where

$$(\widetilde{\mathcal{A}}^{-1})_{kk} = \begin{bmatrix} M^{k,h} + \Delta t\beta\left(K^{k,h} + C^{k,h}(\underline{\mathbf{u}}^{k,h})\right) & \Delta t\beta(D^{k,h})^{\mathrm{T}} \\ \Delta t\beta D^{k,h} & \end{bmatrix}^{-1},$$

$$(\widetilde{\mathcal{B}})_{ik} = \begin{bmatrix} \Delta t\beta B^{[i]k,h\delta} & O \end{bmatrix}.$$

The second equality in Eq. (20) comes from the fact that $\widetilde{\mathcal{A}}$ is block diagonal. Each block of the Schur complement is therefore a sum of the contributions stemming from each subdomain Ω^j. However, we recall that $B^{[i]j,h\delta}$ is not null only if $\Gamma^{[i]}$ is an interface of Ω^j, which typically implies that only a small number of subdomains contribute to each $(\mathcal{S})_{ij}$. As we discussed in Section 4.2, the Lagrange multiplier basis functions required to achieve h-convergence are often very few; hence, the Navier–Stokes matrix inverse (or an approximation thereof) can be efficiently applied to every column of $[B^{[i]k,h\delta}, O]^{\mathrm{T}}$, whose number is typically small.

Remark 5. The special structure of the Schur complement in Eq. (20) makes its computation particularly attractive in the context of high-performance computing, assuming that each subdomain be assigned to one or few processors. An effective partition strategy should take into account the number of degrees of freedom associated with each geometrical building block in order to preserve the load balancing among the set of computational nodes. At the same time, the fact that the coupling conditions do not require to explicitly transfer information from one side of the interface to the other can be exploited to limit the amount of processor communications. In this paper, we only consider a serial implementation, but future directions of the current work include investigations on the parallel performance of the preconditioner.

Remark 6. The computation of the Schur complement $\mathcal{S}$, whose cost scales linearly with the number of columns of $\widetilde{\mathcal{B}}$ (i.e., the total number of Lagrange multiplier basis functions), is in fact the most expensive step of the application of the preconditioner discussed in this section. In our numerical experiments, however, we observed that significant performance improvements are achieved by reusing the same Schur complement for n consecutive applications of the preconditioner. This strategy does not significantly affect the number of GMRES iterations, if n is small enough (in our case, $n \sim 20$).

Let us now address the application of the inverses of $\widetilde{\mathcal{A}}$ and $\mathcal{S}$. Since $\mathcal{S} \in \mathbb{R}^{N_\Gamma N_\lambda \times N_\Gamma N_\lambda}$, the Schur complement is typically small and is inexpensively inverted by solving the linear system either directly or via iterative methods (complemented with standard preconditioners such as multigrid or ILU). Matrix $\widetilde{\mathcal{A}}$ features a block diagonal structure in which each diagonal block is itself a saddle-point system and inverting $\widetilde{\mathcal{A}}$ is therefore equivalent to solving linear systems which are local to each subdomain Ω^j. Employing, instead of $\widetilde{\mathcal{A}}^{-1}$, a suitable approximation thereof, gives rise to different suitable preconditioners.

In this work, we choose to approximate every block diagonal matrix of $\widetilde{\mathcal{A}}^{-1}$ by considering a single application of the SIMPLE preconditioner, both in the computation of the Schur complement shown in Eq. (20) and in the application of the preconditioner. We recall that SIMPLE is based on Eq. (18) applied to the Navier–Stokes equations and that the approximated inverse of the top left diagonal block is performed by extracting and inverting its diagonal [54]. In Fig. 4, we show the robustness of our preconditioner with respect to the number of blocks and to the number of basis functions for the Lagrange multipliers per interface N_λ. The considered geometry is that of the aorta and the illiac arteries in Fig. 4 (left). The blocks are sequentially added starting from the inlet (for this reason, we remark that the size of the system increases proportionally with the number of blocks). We compare the preconditioner performance with that achieved by inverting every block in $\widetilde{\mathcal{A}}$ with GMRES and relatively large tolerances ($5 \cdot 10^{-1}$ and $1 \cdot 10^{-2}$). We remark that, as the preconditioner in the latter approach varies in each iteration, we are obliged to employ flexible GMRES (FGMRES) [56]. If the local systems are solved exactly, the preconditioner is in fact the original global matrix, as Eq. (18) is an exact decomposition. For this reason, solving

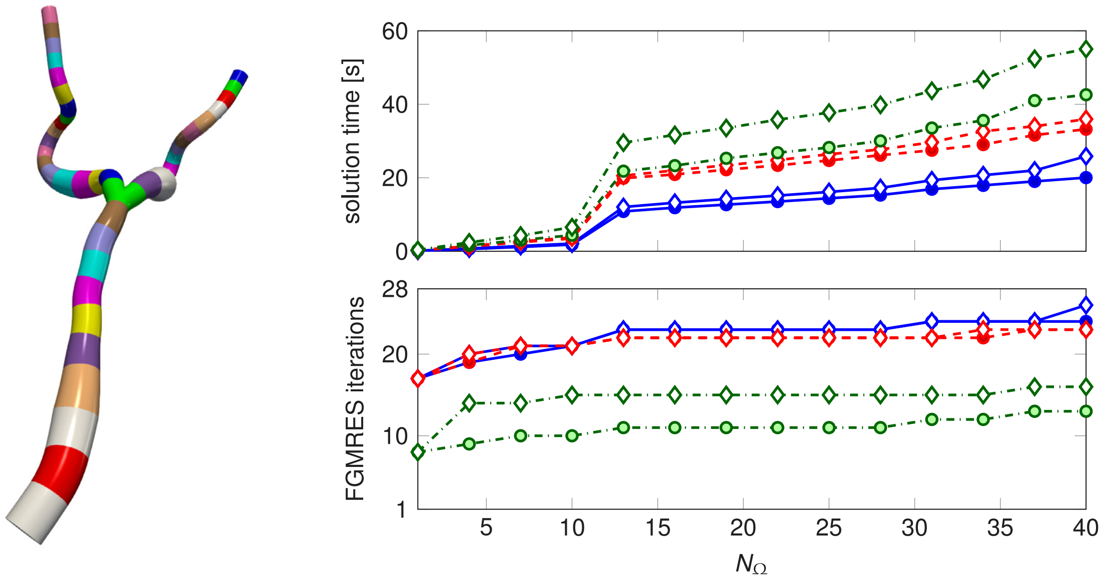

Fig. 4. Solution time of a single linear system preconditioned as discussed in Section 4.3 (right-top) and corresponding number of FGMRES iterations (right-bottom), as functions of number of subdomains N_Ω in the geometry of the aorta and iliac arteries (left). Blue solid lines: $\widetilde{\mathcal{A}}^{-1}$ approximated with a single application of SIMPLE for each subdomain; red dashed lines and blue dash-dotted lines: $\widetilde{\mathcal{A}}^{-1}$ solved with GMRES and tolerance $5 \cdot 10^{-1}$ and $1 \cdot 10^{-2}$, respectively; circles: $N_\lambda = 3$; diamonds = $N_\lambda = 84$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the local linear systems with GMRES leads to a better performance in terms of number of iterations. However, approximating each local inverse with SIMPLE is more efficient in terms of solution time, as each FGMRES iteration is less computational expensive. We conclude by observing that the increase in solution time occurring at $N_\Omega = 13$ is due to the introduction of the bifurcation—which is composed of a larger number of elements than the other blocks—in the set of considered subdomains.

5. The reduced basis element method for flow in arteries

As discussed in Section 4, the subdomains in the target geometry Ω_{m} are obtained from the parametrized geometrical deformation of a number of building blocks $\widehat{\Omega}^i$, $i = 1, \dots, N_{\mathrm{bb}}$. In this paper, these are a model symmetric bifurcation (B), and straight tubes with aspect ratios length/diameter 1:1 (T1), 1:2 (T2) and 1:3 (T3). The offline phase of our reduced order model algorithm consists of defining reduced basis functions in each of these building blocks $\widehat{\Omega}^i$. The snapshots are collected from a single decomposed "artificial" geometry $\Omega_{\mathrm{m}} = \bigcup_{j=1}^{N_\Omega} \Omega^j$ by sampling the geometrical parameters $\underline{\boldsymbol{\mu}}^1, \dots, \underline{\boldsymbol{\mu}}^{N_\Omega}$ describing each subdomain from uniform distributions centered on the values characterizing the original configuration, as depicted in Fig. 5. The snapshots are found by solving a flow problem with $\rho_{\mathrm{f}} = 1.06$ g cm^{-3}, $\mu_{\mathrm{f}} = 4 \cdot 10^{-2}$ g cm^{-1} s^{-1}, the imposed inflow flow rate shown in Fig. 5 (in the box on the left) with a parabolic profile and homogeneous Neumann conditions on the outlets on 165 random configurations of the artificial geometry. There exist other equally valid possibilities to generate the database of snapshots. For example, these could be taken by solving flow problems on a collection of target geometries. This approach allows us to avoid issues related to the random sampling of the geometrical parameters—e.g., physiological feasibility of the resulting global geometry—but requires the aid of an automatic algorithm for the decomposition to be efficient. The development of such an algorithm is one of the possible future extensions of the present work. The simulations are run from $t_0 = 0$ s to $T = 0.3$ s with a BDF scheme of order $\sigma = 2$ and $\Delta t = 2.5 \cdot 10^{-3}$ s. The initial condition at t_0 is computed by gradually increasing the inflow flow rate profile at the inlet by the law $Q(t) = Q_0[1 - \cos((t - t_0^{\mathrm{ramp}})\pi/(t_0 - t_0^{\mathrm{ramp}}))]/2$, Q_0 being the desired flow rate at time t_0, from $t = t_0^{\mathrm{ramp}} = -2 \cdot 10^{-2}$ s to $t = t_0$. For the discretization of the Lagrange multipliers on each interface, we employ the set of basis functions $\widehat{\Xi}_n$ with $n = 5$, which corresponds to $N_\lambda = 63$ basis functions. We remark that the artificial geometry is not included

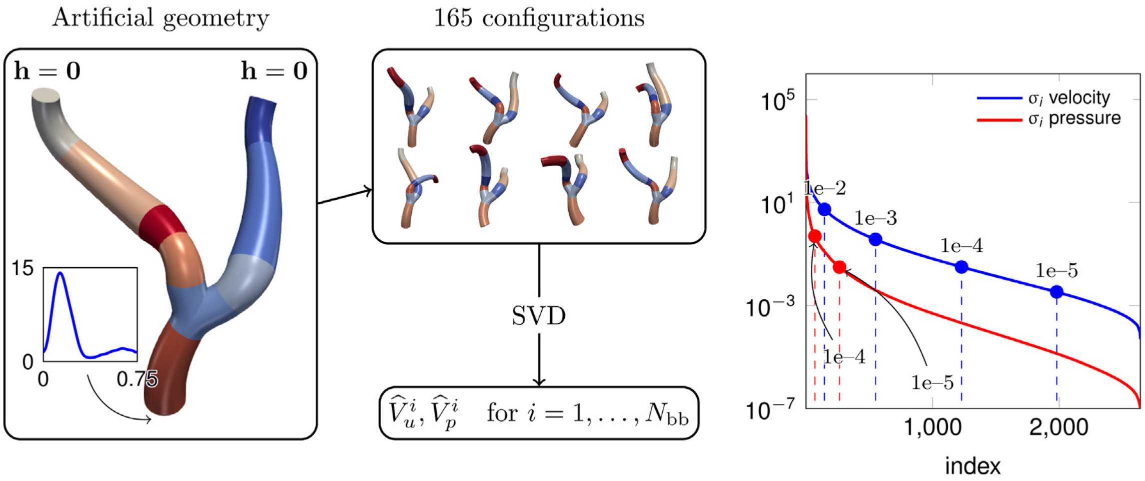

Fig. 5. Offline phase on an artificial geometry featuring $N_\Omega = 9$ subdomains and $N_{\text{bb}} = 4$ building blocks (left) and singular values decay for the velocity and pressure reduced basis built on the bifurcation (right). The snapshots are constructed by solving—on 165 deformed configurations—the flow problem obtained by imposing the flow rate Q depicted in the box on the left (y-axis: Q [cc/s], x-axis: t [s]) at the inlet, and $\mathbf{h} = \mathbf{0}$ on both outflows, in the range $t = (0, 0.3)$. The colored dots on the plot on the right refer to different values of POD tolerances ε_u and ε_p. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in the configurations used for the snapshots generation and is considered in Section 6.1 to assess the performance of the method.

The N_s^i snapshots for velocity and pressure for the ith building block are collected in matrices $\widehat{S}_u^i \in \mathbb{R}^{N_u^{i,h} \times N_s^i}$ and $\widehat{S}_p^i \in \mathbb{R}^{N_p^{i,h} \times N_s^i}$. We remark that, since we are dealing with unsteady problems, these matrices collect snapshots sampled at different timesteps for different values of the geometrical parameters. It is worth noting that each velocity snapshot, which is divergence free in the deformed configuration, does not retain such property on the reference building block. In order to consider snapshots which are divergence-free in the reference configuration, the columns of $\widehat{S}_u^i$ are scaled by means of the divergence-preserving Piola transformation, which is defined as follows. Given a vector field $\mathbf{v}$ defined in $\Omega^j(\underline{\boldsymbol{\mu}}^j)$ and such that $\nabla_{\mathbf{x}} \cdot \mathbf{v} = 0$, the field

$$\widehat{\mathbf{v}}(\widehat{\mathbf{x}}) = |J_{\Phi^{z_j}}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j)| J^{-1}_{\Phi^{z_j}}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j) \mathbf{v}(\Phi^{z_j}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j))$$

is such that $\nabla_{\widehat{\mathbf{x}}} \cdot \widehat{\mathbf{v}} = 0$ in $\widehat{\Omega}^{z_j}$. Matrix $J_{\Phi^{z_j}}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j)$ is the Jacobian of transformation Φ^{z_j} defined in Eq. (9). This explicitly takes the form $J_{\Phi^{z_j}}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j) = (Q(\underline{\boldsymbol{\mu}}^j) J_{\chi^{z_j}}(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}^j))^{-1}$.

Remark 7. The nonaffine deformation $\boldsymbol{\chi}^{z_j}(\cdot; \underline{\boldsymbol{\mu}}^j)$ is defined in analytic form for tubes T1, T2 and T3; therefore, for those building blocks the Jacobian is computed exactly. The nonaffine deformation of the bifurcation B is performed by prescribing the position of the outlets in the physical configuration and by solving a linear elasticity problem such that the displacement field operates the desired rotation of such interfaces. Due to the complications of the evaluation of the Jacobian at the mesh nodes, in the bifurcation we consider $J_{(\chi^{z_j})^{-1}}(\mathbf{x}; \underline{\boldsymbol{\mu}}^j) \approx I$. This simplification is also justified by the fact that for this building block we restrict ourselves to small deformations.

The reduced basis vector for velocity and pressure in the ith building block, $\{\underline{\widehat{\boldsymbol{\zeta}}}_j^{i,h}\}_{j=1}^{N_u^i}$ and $\{\underline{\widehat{\boldsymbol{\eta}}}_j^{i,h}\}_{j=1}^{N_p^i}$, are the columns of matrices

$$\widehat{V}_u^i = [\underline{\widehat{\boldsymbol{\zeta}}}_1^{i,h} | \dots | \underline{\widehat{\boldsymbol{\zeta}}}_{N_u^i}^{i,h}] \in \mathbb{R}^{N_u^{i,h} \times N_u^i}, \qquad \widehat{V}_p^i = [\underline{\widehat{\boldsymbol{\eta}}}_1^{i,h} | \dots | \underline{\widehat{\boldsymbol{\eta}}}_{N_p^i}^{i,h}] \in \mathbb{R}^{N_p^{i,h} \times N_p^i},$$

which are constructed by POD by considering two tolerances ε_u and ε_p (the same for every building block), as described in Section 2.1. Matrices $\widehat{V}_u^i$ and $\widehat{V}_p^i$ are made orthonormal with respect to $\widehat{X}_u^i$ (matrix discretization of the H^1 norm on the ith reference building block) and $\widehat{X}_p^i$ (matrix discretization of the L^2 norm on the ith reference building block) respectively, by following the procedure presented in Remark 1. The first four modes of velocity

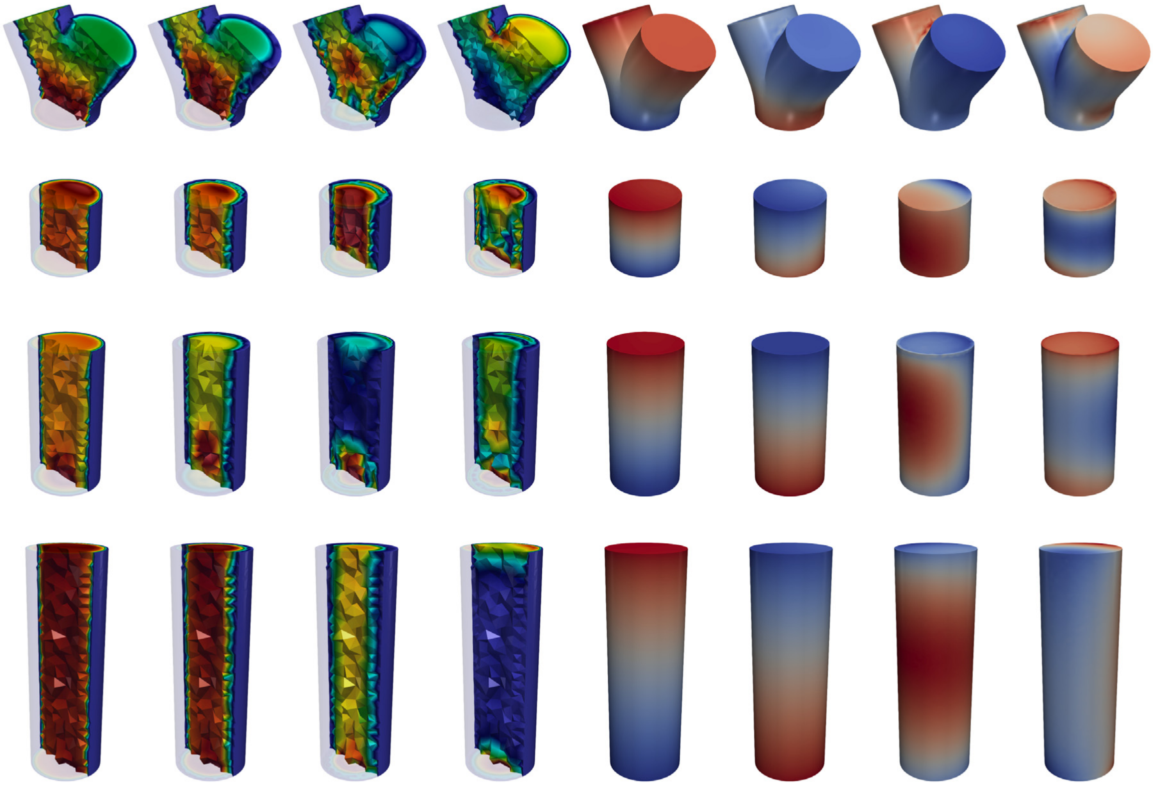

Fig. 6. First four velocity modes of velocity (left) and pressure (right) for the four building blocks. From top to bottom: bifurcation (B), tubes with aspect ratio length/diameter 1:1 (T1), 1:2 (T2) and 1:3 (T3).

and pressure for each building block are depicted in Fig. 6. We also introduce the local matrices V_u^j, which are obtained by applying to each column of $\widehat{V}_u^{z_j}$ the Piola transformation from the reference configuration $\widehat{\Omega}^{z_j}$ to the physical one Ω^j (being dependent on the geometrical parameters $\mathcal{M}$, these must be computed during the online phase). Table 1 reports data about the $N_{\text{bb}} = 4$ building blocks composing the artificial geometry used for the data generation, such as number of snapshots N_s^i and the sizes of FE and RB spaces for velocity and pressure. Although the RB sizes are considerably smaller than the FE ones, the number of basis functions needed to achieve low POD tolerances is substantial. This indicates that the amount of information carried by the snapshots impedes the reduction of the problem. The basis size could be decreased by considering narrower sampling intervals for the geometrical parameters describing each building block. However, as in Section 6.2 we test the ability of the same reduced basis to generalize to the case of a geometry which is not considered in the offline phase, here we decide to allow for significant deviations of the configurations from the original geometry during the snapshots generation. It is worth noting that, in order to decrease the already high computational burden of the offline phase, we settle for a number of configurations (165) that is possibly too limited to capture the geometrical variability we consider in the dataset (see Fig. 5 for examples of some the configurations). As we verify in Section 6.1, the errors that we obtain in the online phase are—although sufficiently low for most cardiovascular applications—considerably larger than the POD tolerance as a consequence of Remark 2.

5.1. Supremizers enrichment for pressure and coupling Lagrange multipliers

In Section 3.2 we recall that the Navier–Stokes equations represent an example of saddle-point equations and that this class of problems is associated with stability issues related with the discretization spaces employed for the primal and dual fields (velocity and pressure, respectively). Unfortunately, even if a stable discretization is considered during the reduced basis generation, the stability is in general not preserved in the reduced system. Furthermore, the

Table 1
Number of snapshots N_s^i, velocity and pressure FE basis sizes $N_u^{i,h}$ and $N_p^{i,h}$, and velocity and pressure RB sizes N_u^i and N_p^i with respect to different POD tolerances ε_u and ε_p, for 4 different building blocks (B: bifurcation, T1, T2 and T3: tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively); $\varepsilon_u^0 = 5 \cdot 10^{-4}$, $\varepsilon_p^0 = 1 \cdot 10^{-5}$.

	N_s^i	$N_u^{i,h}$	$N_p^{i,h}$	N_u^i w.r.t. ε_u						N_p^i w.r.t. ε_p			
				$32\varepsilon_u^0$	$16\varepsilon_u^0$	$8\varepsilon_u^0$	$4\varepsilon_u^0$	$2\varepsilon_u^0$	ε_u^0	$8\varepsilon_p^0$	$4\varepsilon_p^0$	$2\varepsilon_p^0$	ε_p^0
B	2,640	76,974	3,552	99	172	270	394	549	732	82	131	189	265
T1	7,920	42,708	2,162	69	134	233	379	582	848	54	86	133	198
T2	7,920	76,416	3,830	60	103	162	243	354	503	59	92	131	181
T3	5,280	103,728	5,211	23	45	78	131	211	324	29	45	66	95

global system obtained from the nonconforming method introduced in Section 4.2 is also a saddle-point problem where the velocity and the Lagrange multipliers play the role of the primal and dual fields, respectively. Among the ways to deal with the loss of stability in the reduced system are the use of least squares Petrov–Galerkin approaches for the solution of the minimization problem associated to the nonlinear residual of the reduced equations [57,58] and the supremizers enrichment [59–61]. Here we follow the latter approach.

The stability condition is often called inf–sup condition [41] and must be satisfied both at the continuous and discrete level in order to ensure the existence and uniqueness of the respective solutions. Let us consider W2, which we assume to be well-posed in the continuous setting. We first address the stability with respect to the constraint imposed by the pressure (divergence free velocity). At the FE level, the inf–sup condition requires the existence of $\beta_p^{j,h} \in \mathbb{R}$ such that, for all $j = 1, \dots, N_\Omega$,

$$\beta_p^{j,h} = \inf_{\underline{\mathbf{q}} \neq \underline{\mathbf{0}}} \sup_{\underline{\mathbf{v}} \neq \underline{\mathbf{0}}} \frac{\underline{\mathbf{q}}^{\mathrm{T}} D^{j,h} \underline{\mathbf{v}}}{\|\underline{\mathbf{v}}\|_{\mathcal{V}^{j,h}} \|\underline{\mathbf{q}}\|_{Q^{j,h}}} > 0,$$

where we used the notation $\|\underline{\mathbf{v}}\|_{\mathcal{V}^{j,h}} = \underline{\mathbf{v}}^{\mathrm{T}} X_u^{j,h} \underline{\mathbf{v}} = \|\mathbf{v}^h\|_{\mathcal{V}^j}$ and $\|\underline{\mathbf{p}}\|_{Q^{j,h}} = \underline{\mathbf{p}}^{\mathrm{T}} X_p^{j,h} \underline{\mathbf{p}} = \|p^h\|_{Q^j}$. Taylor–Hood elements [42] are an example of stable choice of elements ensuring that $\beta_p^{j,h} > 0$, as mentioned in Section 3.2. In the RB context, we introduce $X_u^{j,N} = (V_u^j)^{\mathrm{T}} X_u^{j,h} V_u^j$, $\|\underline{\mathbf{v}}^N\|_{\mathcal{V}^{j,N}} = \underline{\mathbf{v}}^{\mathrm{T}} X_u^{j,N} \underline{\mathbf{v}}$, the pressure counterparts $X_p^{j,N}$ and $\|\underline{\mathbf{p}}^N\|_{Q^{j,N}}$, and $D^{j,N} = (\widehat{V}_p^{z_j})^{\mathrm{T}} D^j V_u^j$. The inf–sup condition becomes

$$\beta_p^{j,N} = \inf_{\underline{\mathbf{q}} \neq \underline{\mathbf{0}}} \sup_{\underline{\mathbf{v}} \neq \underline{\mathbf{0}}} \frac{\underline{\mathbf{q}}^{\mathrm{T}} D^{j,N} \underline{\mathbf{v}}}{\|\underline{\mathbf{v}}\|_{\mathcal{V}^{j,N}} \|\underline{\mathbf{q}}\|_{Q^{j,N}}} > 0.$$

The main idea of supremizers enrichment is to augment the reduced basis for the velocity with vectors (the supremizers) specifically computed from the pressure modes to ensure the positivity of the inf–sup constant. Formally, let us consider the supremizers $\underline{\mathbf{s}}_1^j, \dots, \underline{\mathbf{s}}_{N_p^{z_j}}^j$ which are obtained by solving for $j = 1, \dots, N_\Omega$ and for $l = 1, \dots, N_p^{z_j}$ the problem $X_u^{j,h} \underline{\mathbf{s}}_l^j = (D^{j,h})^{\mathrm{T}} \underline{\boldsymbol{\eta}}_l^{z_j}$. It can be shown [59] that substituting V_u^j with the enriched basis $\widetilde{V}_u^{j,+} = [V_u^{z_j} | \underline{\mathbf{s}}_1^j | \dots | \underline{\mathbf{s}}_{N_p^{z_j}}^j]$ leads to $\beta_p^{j,N} > \beta_p^{j,h} > 0$. A major drawback of this type of (exact) supremizers enrichment is that the supremizers for subdomain Ω^j are dependent on the geometrical parameter $\underline{\boldsymbol{\mu}}^j$, which implies that they must be computed in the online phase. For this reason, in this paper we follow an approximate approach similar to the one considered in [59] in the case of geometrical parameters. For every reference building block $\widehat{\Omega}^i$ and for $l = 1, \dots, N_p^i$, we introduce the problems $\widehat{X}_u^{i,h} \widehat{\underline{\mathbf{s}}}_l^i = (\widehat{D}^{i,h})^{\mathrm{T}} \widehat{\underline{\boldsymbol{\eta}}}_l^i$, where $\widehat{X}_u^i$ and $\widehat{D}^{i,h}$ are velocity norm and divergence matrices assembled on the ith reference building block. Then, the enriched velocity matrix becomes $\widehat{V}_u^{i,+} = [\widehat{V}_u^i | \widehat{\underline{\mathbf{s}}}_1^i | \dots | \widehat{\underline{\mathbf{s}}}_{N_p^i}^i]$. The coupling stabilization is performed by following the same procedure. In particular, in this case the problems to be solved for each reference building block $\widehat{\Omega}^i$ read $\widehat{X}_u^{i,h} \widehat{\underline{\mathbf{z}}}_l^i = (\widehat{B}^{i[m],h\delta})^{\mathrm{T}} \underline{\mathbf{e}}_l$, for $l = 1, \dots, N_\lambda$; $\widehat{B}^{i[m],h\delta}$ is the matrix assembled on $\widehat{\Omega}^i$ discretizing the coupling with the mth interface (specifically, if $\widehat{\Omega}^i$ is a tube $m = 1, 2$, whereas if it is a bifurcation $m > 2$). In the following, we simply denote by $\widehat{V}_u^i$ the matrix obtained by arranging columnwise $\widehat{\underline{\boldsymbol{\zeta}}}_l^{i,h}$ and the supremizers for the pressure and coupling stabilizations $\widehat{\underline{\mathbf{s}}}_l^i$ and $\widehat{\underline{\mathbf{z}}}_l^i$; the columns of $\widehat{V}_u^i$ are made orthonormal with respect to $\widehat{X}_u^i$ with the Gram–Schmidt algorithm. The numerical results in Section 6 are obtained by following this stabilization strategy.

5.2. Assembly and solution of the global reduced system

Let us define the global matrix

$$\mathcal{W} := \operatorname{diag}\left(\begin{bmatrix} V_u^{z_j} & \\ & \widehat{V}_p^{z_j} \end{bmatrix}\right)_{j=1,\dots,N_\Omega},$$

the matrices

$$\mathcal{M}^N := \mathcal{W}^{\mathrm{T}}\mathcal{M}^h\mathcal{W}, \quad \mathcal{A}^N(\underline{\mathbf{W}}^N) := \mathcal{W}^{\mathrm{T}}\mathcal{A}^h(\mathcal{W}\underline{\mathbf{W}}^N)\mathcal{W}, \quad \mathcal{B}^{N\delta} := \mathcal{B}^{h\delta}\mathcal{W},$$

the vector of reduced degrees of freedom for all the subdomains $\underline{\mathbf{W}}^N = \operatorname{vec}(\underline{\mathbf{w}}^{1,N}, \dots, \underline{\mathbf{w}}^{N_\Omega,N})$, and the vectors encoding the data $\underline{\mathbf{F}}^N := \mathcal{W}^{\mathrm{T}}\underline{\mathbf{F}}^h$ and $\underline{\mathbf{G}}^{N\delta} := \underline{\mathbf{G}}^{h\delta}$. Then, the reduced residual at timestep t_{k+1} is obtained from Eq. (15) and reads

$$\underline{\mathbf{R}}^N(\underline{\mathbf{Y}}_{k+1}^{N\delta}) := \mathcal{H}^N\underline{\mathbf{Y}}_{k+1}^{N\delta} - \sum_{j=1}^{\sigma}\alpha_j\mathcal{H}^N\underline{\mathbf{Y}}_{k-j+1}^{N\delta} - \Delta t\beta\mathring{\underline{\mathbf{F}}}^{N\delta}(t_{k+1}, \underline{\mathbf{Y}}_{k+1}^{N\delta}) = \underline{\mathbf{0}},$$

where

$$\mathcal{H}^N := \begin{bmatrix} \mathcal{M}^N & \\ & \end{bmatrix}, \quad \mathring{\underline{\mathbf{F}}}^{N\delta}(t, \underline{\mathbf{Y}}^N) := \begin{bmatrix} \underline{\mathbf{F}}^N(t) \\ \underline{\mathbf{G}}^{N\delta}(t) \end{bmatrix} - \begin{bmatrix} \mathcal{A}^N(\underline{\mathbf{W}}^N) & (\mathcal{B}^{N\delta})^{\mathrm{T}} \\ \mathcal{B}^{N\delta} & \end{bmatrix}\begin{bmatrix} \underline{\mathbf{W}}^N \\ \underline{\mathbf{\Lambda}}^\delta \end{bmatrix},$$

and $\underline{\mathbf{Y}}^{N\delta} := \operatorname{vec}(\underline{\mathbf{W}}^N, \underline{\mathbf{\Lambda}}^\delta)$.

As discussed in Section 4.3, finding the root of nonlinear equations using the Newton–Raphson algorithm entails the solution of a nonlinear system in the tangent matrix of the corresponding residual. Formally, solving $\underline{\mathbf{R}}^N(\underline{\mathbf{Y}}) = \underline{\mathbf{0}}$ given an initial guess $\underline{\mathbf{Y}}^{(0)}$ leads to the iterative algorithm

$$\underline{\mathbf{Y}}^{(l+1)} = \underline{\mathbf{Y}}^{(l)} - \left(\mathcal{J}_{\underline{\mathbf{R}}^N}\left(\underline{\mathbf{Y}}^{(l),N}\right)\right)^{-1}\underline{\mathbf{R}}^N\left(\underline{\mathbf{Y}}^{(l)}\right),$$

which is equivalent to Eq. (16) in the reduced context.

The efficiency of the reduction relies on the fast assembly of the tangent matrix $\mathcal{J}_{\underline{\mathbf{R}}^N}$ and residual $\underline{\mathbf{R}}^N$. Regarding the former, we consider the following approximation

$$\widetilde{\mathcal{J}}_{\underline{\mathbf{R}}}^N = \begin{bmatrix} \mathcal{M}^N + \Delta t\beta\mathcal{A}_{\mathrm{lin}}^N & \Delta t\beta(\mathcal{B}^{N\delta})^{\mathrm{T}} \\ \Delta t\beta\mathcal{B}^{N\delta} & \end{bmatrix}, \tag{21}$$

where

$$\mathcal{A}_{\mathrm{lin}}^N := \mathcal{W}^{\mathrm{T}}A_{\mathrm{lin}}\mathcal{W}, \quad \mathcal{A}_{\mathrm{lin}} := \operatorname{diag}\left(\begin{bmatrix} K^{j,h} & (D^{j,h})^{\mathrm{T}} \\ D^{j,h} & \end{bmatrix}\right)_{j=1,\dots,N_\Omega}$$

is the matrix obtained by neglecting the convective terms in the Navier–Stokes equations. We remark that the reduced tangent matrix features a saddle-point structure as its full order counterpart in Eq. (17). Therefore, system $\widetilde{\mathcal{J}}_{\underline{\mathbf{R}}^N}\underline{\mathbf{X}}^N = \underline{\mathbf{B}}^N$ can be solved directly by applying the reduced version of Eq. (19). The advantages of this approach are: (i) the tangent matrix $\widetilde{\mathcal{J}}_{\underline{\mathbf{R}}^N}$ is never entirely allocated, because every stage for applying Eq. (19) involves operations that are local to either subdomains or interfaces (we recall that inverting $\mathcal{A}_{\mathrm{lin}}$ amounts to inverting each of its diagonal blocks), (ii) as a result of approximation (21), the tangent matrix is the same for every solution of the linear system; hence the reduced Schur complement is assembled only once and it can be factorized—along with the other local matrices to be inverted—at the start of the simulation. The linearized version of the tangent matrix (21) is nonconsistent, which implies that the Newton–Raphson algorithm is not expected to convergence quadratically. However, the reduced complexity of the assembly results in a considerable performance gain overall.

The problematic part of the computation of $\underline{\mathbf{R}}^N(\underline{\mathbf{Y}})$ is evidently the nonlinear term, as the matrices encoding the linear ones are computed only once and the corresponding contributions are found at each timestep by inexpensive matrix–vector multiplications. After trivial but repetitious steps, we find that the blocks of the nonlinear part of the reduced residual read, for all $j = 1, \dots, N_\Omega$,

$$\underline{\mathbf{c}}^{j,N} = \left(V_u^{z_j}\right)^{\mathrm{T}} C^{j,h}\left(V_u^{z_j}\underline{\mathbf{u}}^{j,N}\right) V_u^{z_j}\underline{\mathbf{u}}^{j,N}. \tag{22}$$

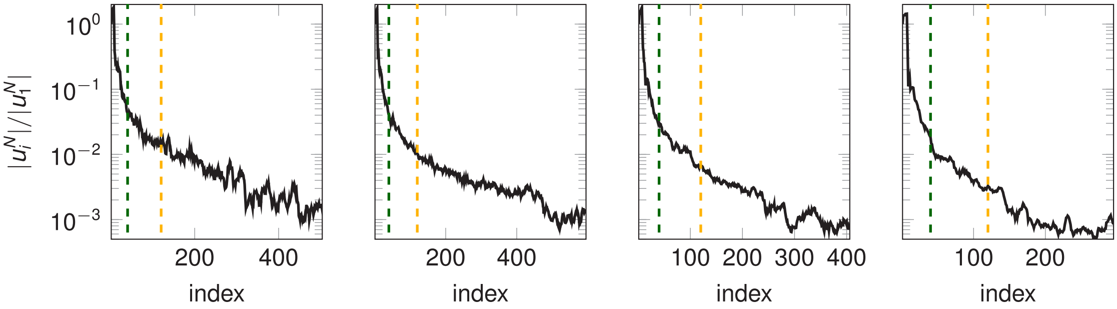

Fig. 7. Average over time of the RB velocity solutions—normalized with respect to the first coefficient—in the four building blocks (from left to right: bifurcation B and tubes T1, T2 and T3) in the RB simulation considered in Section 6.1 with $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. The left and right dashed lines in every plot correspond to the indices 40 and 120, which are two of the choices considered in Section 6.1 for the truncation of the computation of the convective term.

One way to compute the nonlinear term for every block is then to assemble the full order nonlinear term $C(V_u^{z_j} \underline{\mathbf{u}}^{j,N}) V_u^{z_j} \underline{\mathbf{u}}^{j,N}$ and to project it onto the reduced space. Another strategy is based on the decomposition

$$\underline{\mathbf{c}}_i^{j,N} = \sum_{l=1}^{N_u^{z_j}} \sum_{m=1}^{N_u^{z_j}} u_l^{j,N} u_m^{j,N} \int_{\Omega^j} \left[\left(\boldsymbol{\zeta}_m^{z_j,h} \cdot \nabla \right) \boldsymbol{\zeta}_l^{z_j,h} \right] \boldsymbol{\zeta}_i^{z_j,h}. \tag{23}$$

The integrals in Eq. (23) are independent of the reduced solution and can be computed as a setup step in the first stages of the simulation. However, the amount of computation increases quadratically with the size of the reduced basis and could therefore nullify the performance gain. Using the fact that the velocity modes in $V_u^{z_j}$ are sorted in order of significance as a consequence of Proposition 1, it is legitimate to consider the following approximation

$$\underline{\mathbf{c}}_i^{j,N} \approx \sum_{l=1}^{N_c^{z_j}} \sum_{m=1}^{N_c^{z_j}} u_l^{j,N} u_m^{j,N} \int_{\Omega^j} \left[\left(\boldsymbol{\zeta}_m^{z_j,h} \cdot \nabla \right) \boldsymbol{\zeta}_l^{z_j,h} \right] \boldsymbol{\zeta}_i^{z_j,h}, \tag{24}$$

where $0 < N_c^j \leq N_u^{z_j}$. In other words, the observation that the magnitude of the reduced coefficients $u_i^{j,N}$ quickly decreases as i increases—as we show in Fig. 7 for the case of one of the simulations presented in Section 6.1—allows to truncate the two sums in Eq. (23) to the first N_c^j terms. In the numerical results in Section 6 we investigate the effects of considering both Eqs. (22) and (24) for the computation of the convective part of the residual.

6. Numerical results

In this section, we assess the performance of our numerical method in two applications. In Section 6.1, we consider the same modular artificial geometry employed in the offline phase presented in Section 5 and we compare the results obtained by solving the flow problem using the RB and the FE methods. These are obtained with a code based on LifeV, a C++ FE library with support to high-performance computing [62]. In Section 6.2, we consider a simple but more physiological geometry of an aorta and the two iliac arteries. In this case, we compare the results obtained with the RB method against the ones obtained on a reference geometry (i.e., not partitioned into approximated subdomains) with SimVascular[1] [63], an open-source software for patient-specific modeling and blood flow simulations.

For all the simulations, we fix $\rho_f = 1.06$ g cm^{-3}, $\mu_f = 4 \cdot 10^{-2}$ g cm^{-1} s^{-1}, and we consider the same choice for the discrete Lagrange multipliers space as in the snapshot generation phase (i.e., $N_\lambda = 63$ for each interface).

6.1. Online phase on the artificial geometry

We evaluate the performance of our reduced order model on the artificial problem employed for the generation of the reduced basis. We recall that the *Artificial geometry* in Fig. 5, which is not included in the set of 165

[1] http://simvascular.github.io/.

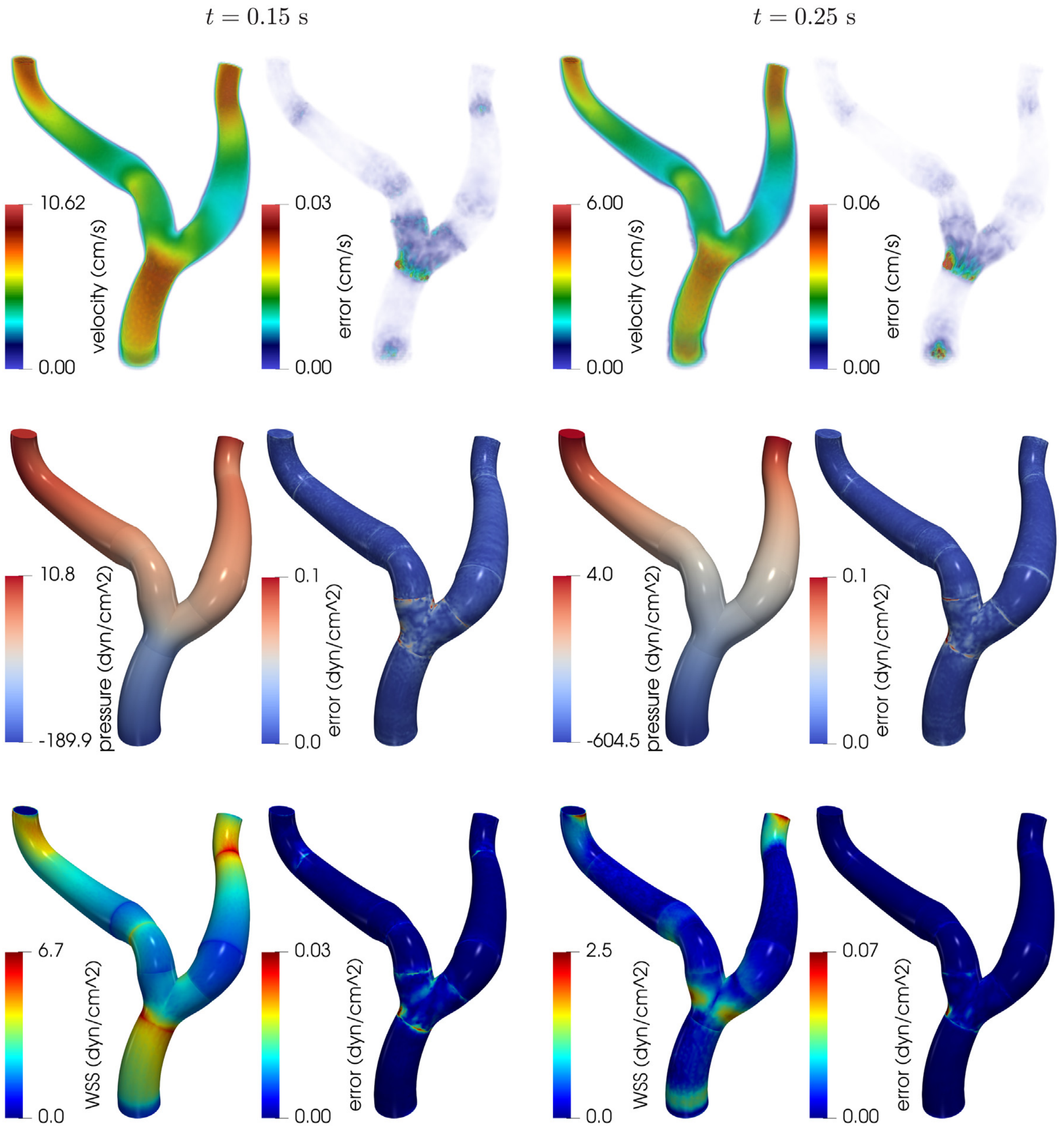

Fig. 8. The left and right columns—each composed of two sub-columns of plots—refer to time $t = 0.15$ s and $t = 0.25$ s, respectively. First row: velocity magnitude volume plot of the RB solution (sub-column left) and magnitude of the point-wise velocity error w.r.t. the FE solution (sub-column right). Second row: pressure plot of the RB solution (sub-column left) and absolute value of the point-wise pressure error w.r.t. the FE solution (sub-column right). Third row: magnitude of the WSS of the RB solution (sub-column left) and magnitude of the point-wise WSS error w.r.t. the FE solution (sub-column right). The RB solution corresponds to the choice $\varepsilon_u = 1 \cdot 10^{-3}$ and $\varepsilon_p = 1 \cdot 10^{-5}$.

configurations used to produce the snapshots, is a legitimate candidate to test the accuracy on geometries not “seen” in the offline phase. The solution by the RB method is compared to the global solution obtained by considering FE method solutions in each subdomain (with the same meshes used in the RB case) coupled with the discretization strategy presented in Section 4.2. We consider the same choice for the discrete Lagrange multipliers space as in the snapshot generation phase. The reasons for considering such comparison are the following: (i) being that the geometry is exactly the same, it is possible to easily compute H^1 and L^2 errors for velocity and pressure in order to verify the convergence of the RB approximation with respect to the FE one, and (ii) it is possible to fairly

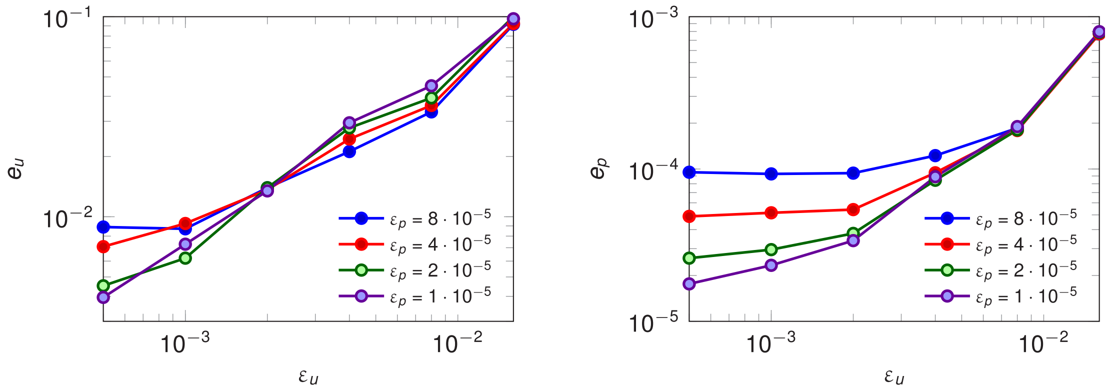

Fig. 9. Error in velocity e_u (left) and error in pressure e_p (right), computed as in Eq. (25), in function of the POD tolerances for velocity and pressure ε_u and ε_p.

Table 2
Overall speedups w.r.t. the FE solution and, in parenthesis, speedups of the solve part of the online phase, i.e., speedup relative to the total running time excluding the setup part in which the reduced bases are loaded and the constant matrices are assembled and projected onto the reduced spaces.

$\varepsilon_p \backslash \varepsilon_u$	$1.6 \cdot 10^{-2}$	$8 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
$8 \cdot 10^{-5}$	33(50)	30(48)	26(43)	22(39)	17(32)	14(31)
$4 \cdot 10^{-5}$	27(46)	25(44)	23(38)	21(38)	17(35)	13(30)
$2 \cdot 10^{-5}$	28(45)	26(43)	23(41)	28(38)	15(33)	12(28)
$1 \cdot 10^{-5}$	26(45)	25(44)	22(41)	17(33)	14(30)	12(29)

discuss the speedup achieved by the RB method, as the RB and FE solutions share the same computational mesh. As for the generation of the reduced basis, we consider $t_0 = 0$ s, $T = 0.3$ s, and a second order BDF scheme with $\Delta t = 2.5 \cdot 10^{-3}$ s.

Fig. 8 shows, in the first two rows, the magnitude of the velocity field and pressure distribution at times $t = 0.15$ s and $t = 0.25$ s obtained with the RB method and the corresponding point-wise errors with respect to the FE solution. The POD tolerances in every subdomain have been set to $\varepsilon_u = 1 \cdot 10^{-3}$ and $\varepsilon_p = 1 \cdot 10^{-5}$. We observe that, despite the global mesh being nonconforming, the velocity and pressure appear to be quite smooth at the interfaces. The comparison with the FE solution highlights the fact that the largest errors are committed in the region of the bifurcation. This is likely due to the fact that, as shown in Table 1, the reduced basis for the corresponding building block (B) is based on a smaller number of snapshots. However, the RB and the FE solutions match quite accurately overall, as the relative error is negligible in every part of the domain. The last row of Fig. 8 depicts the distribution of the magnitude of the WSS on the boundary of the artery in the RB and the magnitude of the error. The influence of the coupling is noticeable: indeed, it is clearly possible to spot the interfaces as regions with abnormally low or high WSS. However, this effect is not due to the RB approximation but rather to the coupling strategy: indeed, the RB and FE approximations are extremely close, as proven by the small magnitude of the error in the WSS.

Fig. 9 shows the H^1 and L^2 relative errors on velocity and pressure integrated in time, defined as

$$e_u^2 := \frac{\int_0^T \|\mathbf{u}^{(j),h} - \mathbf{u}^{(j),N}\|_{b,\mathcal{V}}^2}{\int_0^T \|\mathbf{u}^{(j),h}\|_{b,\mathcal{V}}^2}, \quad e_p^2 := \frac{\int_0^T \|p^{(j),h} - p^{(j),N}\|_{b,\mathcal{Q}}^2}{\int_0^T \|p^{(j),h}\|_{b,\mathcal{Q}}^2}, \tag{25}$$

where $\|\mathbf{u}\|_{b,\mathcal{V}}^2 = \sum_{i=1}^{N_\Omega} \|\mathbf{u}^{(j)}\|_{\mathcal{V}^{(j)}}^2$ and $\|p\|_{b,\mathcal{Q}}^2 = \sum_{i=1}^{N_\Omega} \|p^{(j)}\|_{\mathcal{Q}^{(j)}}^2$ are the broken norms. The errors are plotted as functions of the velocity and pressure POD tolerances ε_u and ε_p, highlighting the convergence of the RB solution to the FE one as the reduced basis size increases. Clearly, ε_u and ε_p both contribute to the errors in velocity and pressure. Indeed, for large ε_p, e_u and e_p set on a plateau as ε_u decreases, indicated that the error in the pressure is dominating the error in the velocity. For each data point in Fig. 9, the corresponding speedup is reported in Table 2. The runtime of the reference FE solution—which is composed of 641,502 degrees of freedom for velocity and pressure and 567 degrees of freedom for the Lagrange multipliers—is 66,892 s (∼ 18.5 h). The speedups are relative to the total runtime and, in parentheses, to the part of the online phase after the initial setup (which include the loading of the reduced basis, the assembly of the constant matrices and their projection onto the reduced spaces). The motivations to consider both speedups are twofold. Firstly, in this paper we do not focus on the optimization of the assembly part of the system, which could considerably increase the total speedup; such optimization could be carried out, for example, by employing (M)DEIM, as mentioned in Section 2.2. Secondly, the setup part of the RB algorithm is particular to the geometry we are interested in. As a matter of fact, should we be interested in solving flow problems corresponding to different boundary conditions and/or fluid properties but on the same geometry, the setup phase can be executed only once, and for each solution of the reduced system we take advantage of the speedups relative to the only solve phase. The gain in performance is in all cases quite substantial (at least one order of magnitude with respect to the full order solution), and we observe, as expected, the trend of increasing speedup as the size of the reduced system decreases. However, the careful profiling of the simulation highlights that most of the time of the solve phase is spent in the assembly of the reduced convective term rather than in the actual solution of the reduced system. This is because, as discussed in Section 5.2, the exact assembly of the reduced convective terms entails two projections and the construction of the full order convective term.

With the purpose of achieving higher speedups during the solution time, we consider the approximation of the convective term given in Eq. (24). Fig. 10 shows the absolute and relative H^1 and L^2 errors on velocity and pressure over time in function of different degrees of truncation of the convective term (i.e., different values of $N_c^{z_j}$, which we set equal to $N_c^{z_j} = N_c$ for every subdomain). The achieved speedups are, from $N_c = 10$ to $N_c = 120$ and using the same notation adopted in Table 2, 56(998), 36(620), 22(464), 10(313), 5(215). The POD tolerances are constant and take the values $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. We remark that the values of N_c are to be considered in relation with the decay of the RB solutions shown in Fig. 7, which refer to the velocity coefficients of the reference solution corresponding to $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. As expected, the runtime of the solve phase is greatly decreased with respect to both the FE solution, against which the speedup achieved is always higher than 200, and with respect to the RB solution with the convective term computed as in Eq. (22). As N_c increases, the total speedup rapidly decreases due to the quadratic dependence on that parameter of the number of integrals computed during the setup phase. Nevertheless, we believe that this strategy for approximating the convective term is of great benefit whenever it is required to run multiple simulations on the same geometry, as in this scenario the setup phase is only performed once.

6.2. *Online phase on the aorta and iliac arteries*

In this section we consider a physiological geometry of an aorta with the two iliac arteries.[2] Our goal is to evaluate the effects of the geometrical approximation on the solution given by our ROM. In order to do so, we employ the geometries depicted in Fig. 11. Specifically, on the left we show the decomposed geometry along with the "exact" one. On the right, we provide a quantitative analysis of the difference between the two. The algorithm to generate the decomposed geometry from the target one is out of the scope of this paper. However, the development of efficient and accurate reconstruction strategies is a topic of interest and will be addressed in future works. Importantly, we choose to employ the same set of RB basis functions computed in the offline phase described in Section 5 (which, we recall, is built upon modifications of the same artificial geometry we use to test the accuracy of the ROM in the previous section). This is motivated by the perspective of employing the method in realistic scenarios which may be considerably different from the ones explored during the offline phase.

The flow problem consists of imposing the same inflow profile shown in Fig. 5 at the inlet (the aorta) and homogeneous Neumann conditions to the outlets (the iliac arteries). We take $T = 1.5\,\mathrm{s}$ (i.e., two heartbeats) and $\Delta t = 1.25 \cdot 10^{-3}\,\mathrm{s}$. As already anticipated, the reference simulation is computed with the SimVascular solver svSolver. This software is based on the FE method with P1-P1 elements and VMS-SUPG stabilization; we refer to [64] for more information regarding this numerical approach. It is therefore challenging to devise a *fair* comparison between the ROM—which, we recall, is built upon a P2-P1 discretization—and the reference solution in terms of efficiency and accuracy. Nevertheless, we provide for sake of completeness some data regarding the reference solution. This is computed on a fine mesh (composed of 1,823,827 nodes) which is selected by studying the

[2] A SimVascular tutorial based on the same geometry considered here is available on the software website (http://simvascular.github.io/docsQuickGuide.html).

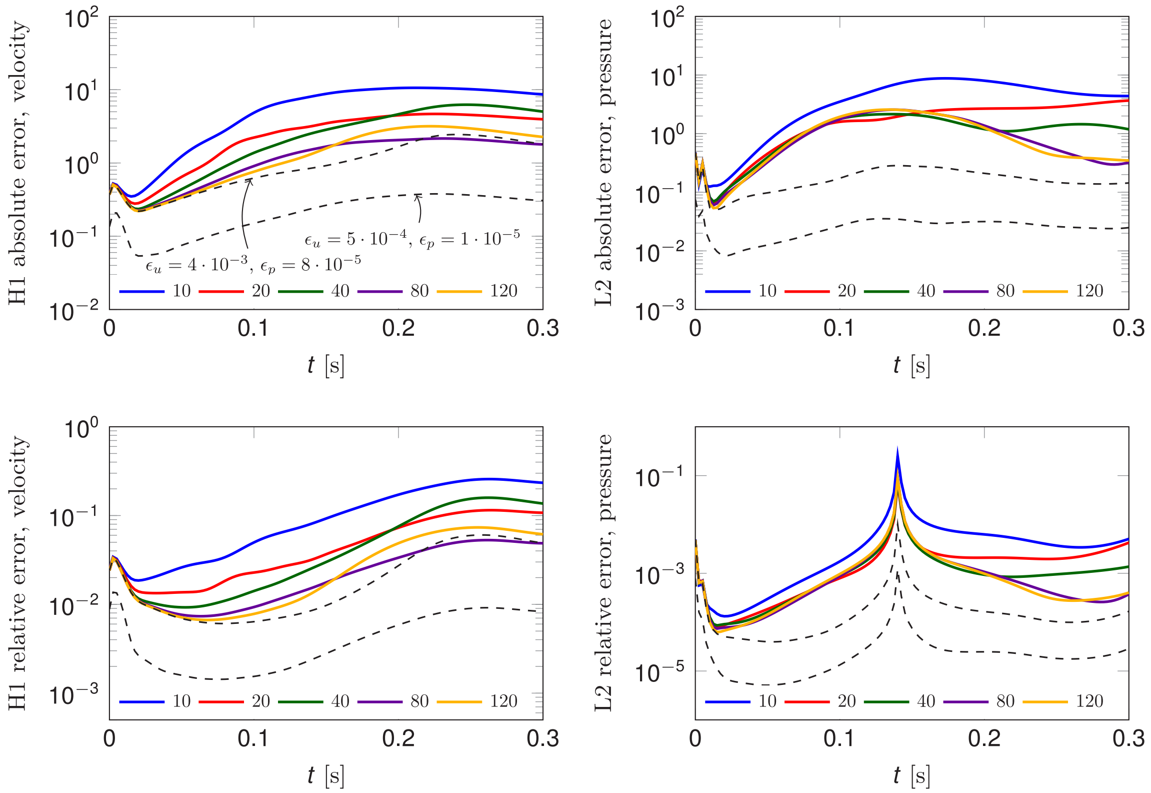

Fig. 10. Errors of reduced velocity and pressure against the FE solution vs time. The colored lines refer to different choices of N_c for the approximation of the nonlinear convective term. The black dashed lines show the reference errors obtained without approximation of the convective term with $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$ (which are the same tolerances used in the simulations corresponding to the colored lines) and $\varepsilon_u = 1 \cdot 10^{-3}$ and $\varepsilon_p = 1 \cdot 10^{-5}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

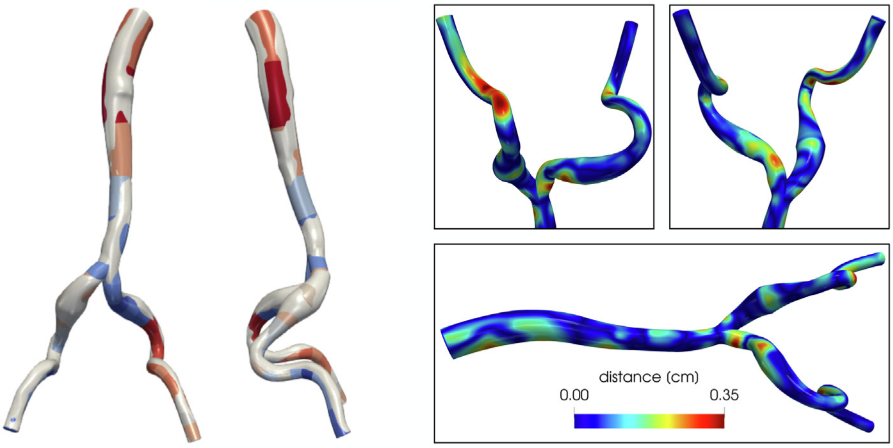

Fig. 11. On the left, qualitative comparison of the reference mesh with the decomposed one. On the right, quantitative estimation of the distance between the two.

convergence of the WSS on the boundary; the simulation using SimVascular took 46,457 s (∼ 13 h) by employing 28 cores.

In Fig. 12, we show the qualitative comparison of the velocity field magnitude, pressure and WSS distribution on the wall at two different timesteps. The RB solution is obtained with $\varepsilon_u = 1 \cdot 10^{-3}$ and $\varepsilon_p = 1 \cdot 10^{-5}$. We observe that, despite the differences in the employed geometries and in the underlying numerical discretization, the solutions share similar features. For instance, the pressure distribution is qualitatively almost identical, and the ranges for velocity and WSS magnitude achieved in every region are comparable. It is apparent, however, that most of the error (on the velocity magnitude in particular) is in the vicinity of the bifurcation. This is due to the fact that our choice of geometrical parameters for the corresponding building block does not allow for the reference bifurcation to be deformed into the target one with sufficient accuracy. For example, Fig. 2 shows that we do not take into account the possibility of varying the radii of the outlets, and this reflects in a large geometric error particularly on one of the branches, as depicted in Fig. 11 (bottom branch in the bottom right plot).

A more quantitative analysis of the performance of the ROM with respect to the reference solution is presented in Fig. 13. Here, we show the average of the WSS magnitude over the three regions highlighted in the figure on the left, the pressure at the inlet and the outflow rate at the outlets, for the reference solution and for RB solutions corresponding to different choices of tolerances and truncations for the approximation of the nonlinear term. We chose to focus on a "fine" RB solution (RB1), where we do not apply the truncation of the convective term, and more "coarse" but efficient RB solutions with approximation of the convective term (RB2, RB3, RB4); for details regarding the employed POD tolerances and number of terms in the truncated sum, we refer the reader to the caption of Fig. 13. The setup, solve and total runtimes in seconds for these simulations—which are run, differently from the reference simulation, on a single core—are the following: for RB1, $4{,}240 + 36{,}170 = 40{,}410$ ($\sim$ 11 h, speedup of 1.1), for RB2, $19{,}382 + 4{,}760 = 24{,}142$ ($\sim$ 6.5 h, speedup of 2), for RB2, $7{,}456 + 2{,}398 = 9{,}854$ ($\sim$ 2.5 h, speedup of 5), for RB3, $1{,}816 + 3{,}774 = 5{,}590$ ($\sim$ 1.5 h, speedup of 8). In all cases, we achieve speedups larger than one with respect to the reference simulation obtained with SimVascular (although the gain is negligible in the case of RB1) but on a single core instead of 28. From the results presented in Fig. 13, we note that, while the approximation of the pressure and flow rate is extremely precise for all RB settings, the performance on the WSS is more challenging. The curves for the average WSS on the two regions on the iliac arteries (B and C) are quite close to the reference one compared to the average WSS on the bifurcation (A). However, this is likely an effect of the geometric approximation rather than the accuracy of the ROM per se. As a matter of fact, we already noted in Fig. 12 that the largest errors are located in that area. We also remark that the POD tolerance plays a more dramatic role in the quality of the solution than the number of terms retained in the truncated nonlinear term N_c. Indeed, the simulations with smallest tolerances (RB1 and RB2) and largest ones (RB3 and RB4) lead to similar results, regardless of the value of N_c. Nevertheless, truncating the convective term is beneficial to the efficiency of the ROM.

7. Conclusions

In this paper, we presented an implementation of the reduced basis element method for the solution of the unsteady 3D Navier–Stokes equations in the context of cardiovascular simulations. We first considered the problem of coupling finite element solutions defined on subdomains obtained from parametrized geometrical deformations of reference building blocks. This was necessary, as the offline phase of our reduced order method requires the generation of snapshots from coupled finite element flow solutions obtained on a variety of geometries. In order to improve the efficiency of the coupled finite element solver, we devised an ad-hoc preconditioner which takes advantage of the saddle-point structure of the discretized linear system. In the following parts of the paper, we formulated the reduced order model by projecting the matrices and variables (velocity and pressure) onto the reduced basis spaces. This procedure is beneficial because it allows us to considerably reduce the number of degrees of freedom (hence, the size of the linear system to be solved in each iteration of the Newton–Raphson algorithm). In the numerical simulations, we demonstrated the capabilities of the method on the same geometry used for the offline phase and on a physiological geometry consisting of an aorta with the two iliac arteries. In the first case, we registered considerable speedups (from 12 to 33 over the total runtime and from 29 to 50 over the sole solve phase) with respect to the full order solution. Considerable gain in performance was also achieved in the second case for some choices of POD tolerances, although the fact that we considered a reference solution obtained with a different solver (i.e., SimVascular) made the comparison in terms of runtime more complex. In both applications we also analyzed the performance in terms of wall-shear stress reconstruction, which is possible in our reduced order method—as opposed, for example, to 1D models—because the 3D nature of the flow problem is preserved.

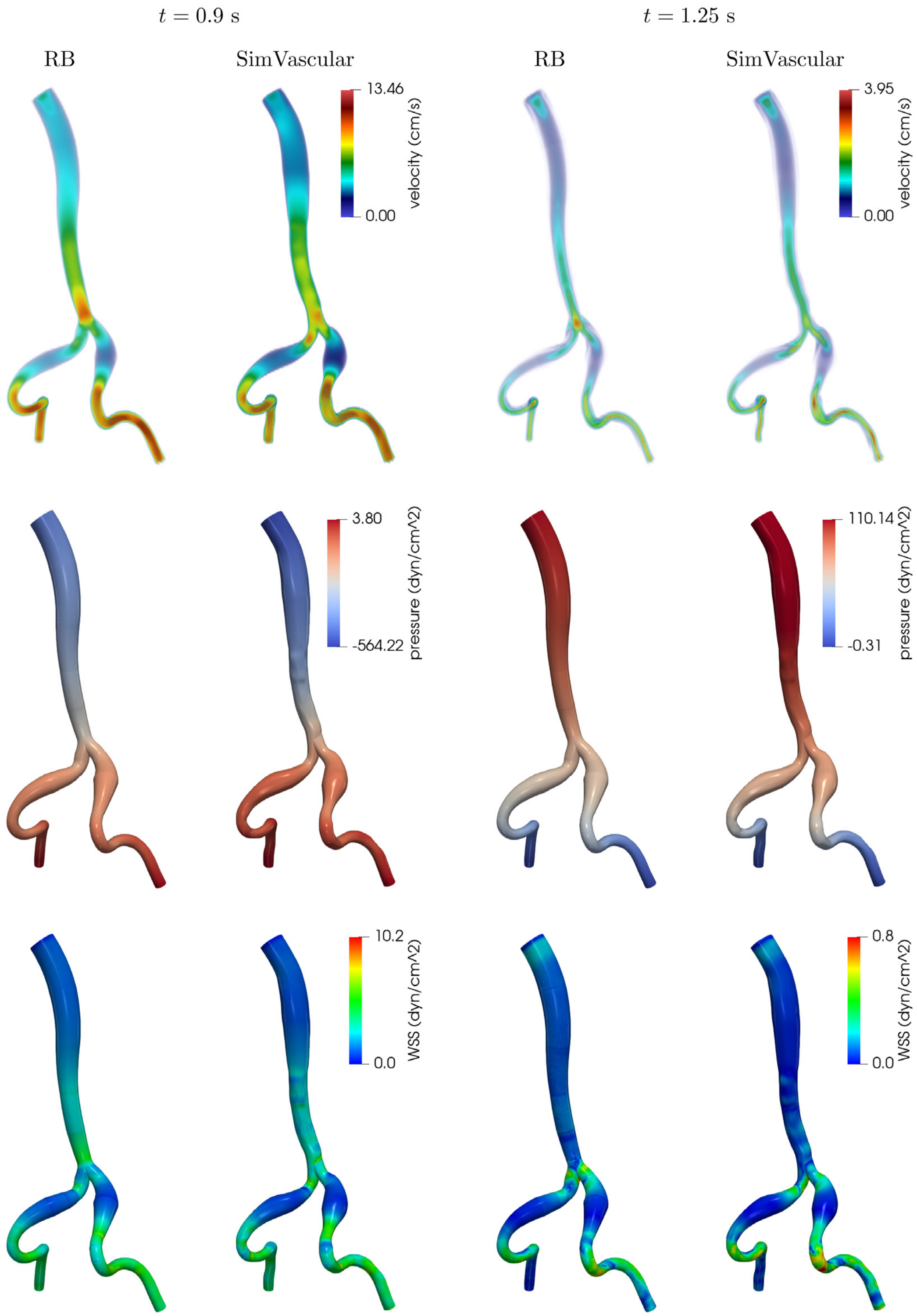

Fig. 12. The left and right columns—each composed of two sub-columns of plots—refer to time $t = 0.9$ s and $t = 1.25$ s, respectively. First row: velocity magnitude volume plot of the RB and reference solutions (sub-column left and sub-column right, respectively). Second row: pressure plot of the RB and reference solutions (sub-column left and sub-column right, respectively). Third row: magnitude of the WSS of the RB solution and reference solution (sub-column left and sub-column right, respectively). The RB solution corresponds to the choice $\varepsilon_u = 1 \cdot 10{-}3$ and $\varepsilon_p = 1 \cdot 10^{-5}$.

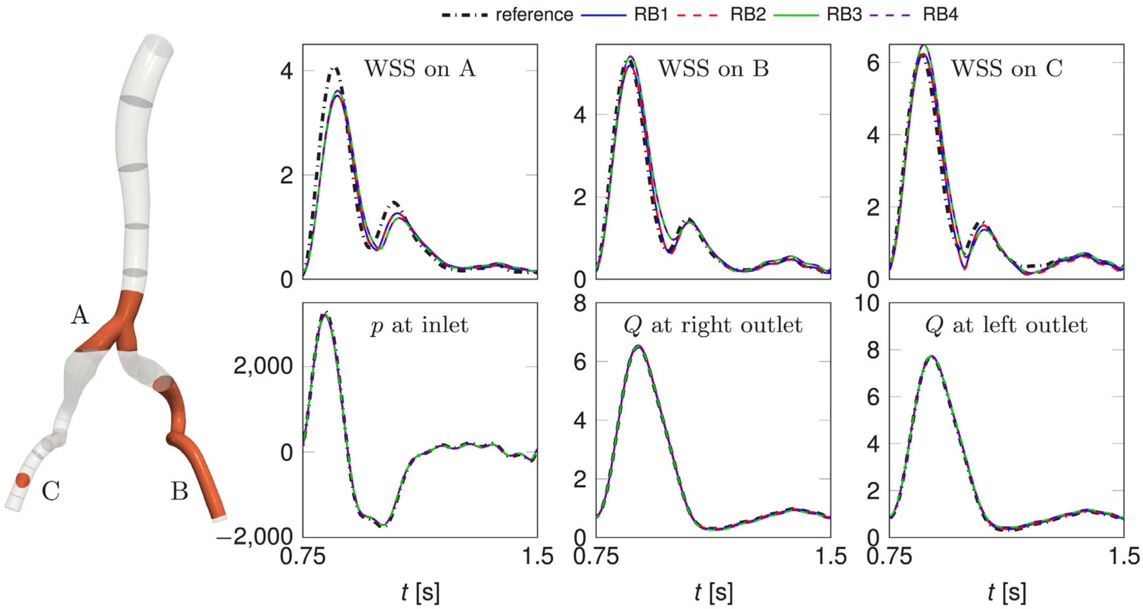

Fig. 13. Average WSS (in dyn/cm^2) on the three regions marked on the figure on the left (top row), and pressure p (in dyn/cm^2) and flow rates Q (in cc/s) at inlet and outlets, respectively (bottom row). The black dashed line refers to the reference solution computed by SimVascular, whereas the 4 colored lines are obtained with different RB settings. RB1: $\varepsilon_u = 4 \cdot 10^{-3}$, RB2: $\varepsilon_u = 8 \cdot 10{-}3$ and $N_c = 80$, RB3: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 40$, RB4: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 20$. In all cases, $\varepsilon_p = 1 \cdot 10^{-5}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We believe that the results presented in this work are promising and that our study suggests many possibilities for the future developments of this reduced order method. As we discuss more in depth in the relative sections of this paper, these include: (i) a parallel implementation of the devised saddle-point preconditioner which could exploit the special structure of the linear system (see Remark 5), (ii) an offline strategy based on the solution of physiological flow problems (rather than flow problems defined on artificially deformed geometries), (iii) efficient ways to reduce the complexity of the setup phase and the assembly of the reduced convective term, and (iv) automatic algorithms for the generation of accurate decomposed geometries out of medical images or reference meshes. Regarding this last point, we also think that—due to the geometrical difficulties to map reference bifurcations into target ones (see discussion in Section 6.2)—an interesting follow-up of the current work is the study of a hybrid method in which some parts of the target geometry (e.g., the bifurcations) are modeled by means of the finite element method and others by means of the reduced basis method. This approach would allow us to easily treat cases featuring more complex geometries—for instance, cerebral aneurysms—that are not approximated as trivial deformations of tubes and which are currently too challenging to tackle with a strategy purely based on the reduced basis method.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research of Luca Pegolotti and Simone Deparis was supported by the Swiss National Science Foundation (SNSF), projects No. 188031 and No. 140184. The work of Alison Marsden and Martin Pfaller was supported by the National Institutes of Health (NIH), grant 1R01LM013120. The authors gratefully acknowledge the Scientific IT and Application Support at EPFL for providing the computational resources necessary to our numerical simulations and the reviewers, who contributed to ameliorating this paper with their feedback.

References

[1] G. Bao, Y. Bazilevs, J.-H. Chung, P. Decuzzi, H.D. Espinosa, M. Ferrari, H. Gao, S.S. Hossain, T.J.R. Hughes, R.D. Kamm, et al., USNCTAM perspectives on mechanics in medicine, J. R. Soc. Interface 11 (97) (2014) 20140301.

[2] C.A. Figueroa, C.A. Taylor, A.L. Marsden, Blood Flow, American Cancer Society, 2017, pp. 1–31.

[3] S. Sankaran, H.J. Kim, G. Choi, C.A. Taylor, Uncertainty quantification in coronary blood flow simulations: impact of geometry, boundary conditions and blood viscosity, J. Biomech. 49 (12) (2016) 2540–2547.

[4] S. Sankaran, A.L. Marsden, A stochastic collocation method for uncertainty quantification and propagation in cardiovascular simulations, J. Biomech. Eng. 133 (3) (2011).

[5] P. Chen, A. Quarteroni, G. Rozza, Simulation-based uncertainty quantification of human arterial network hemodynamics, Int. J. Numer. Methods Biomed. Eng. 29 (6) (2013) 698–721.

[6] A.C.I. Malossi, J. Bonnemain, Numerical comparison and calibration of geometrical multiscale models for the simulation of arterial flows, Cardiovasc. Eng. Technol. 4 (4) (2013) 440–463.

[7] A.C.I. Malossi, Partitioned Solution of Geometrical Multiscale Problems for the Cardiovascular System: Models, Algorithms, and Applications (Ph.D. thesis, Ph. D. thesis), École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2012.

[8] A.L. Marsden, M. Esmaily-Moghadam, Multiscale modeling of cardiovascular flows for clinical decision support, Appl. Mech. Rev. 67 (3) (2015) 030804.

[9] P.J. Blanco, C.A. Bulant, L.O. Müller, G.D.M. Talou, C.G. Bezerra, P.A. Lemos, R.A. Feijóo, Comparison of 1D and 3D models for the estimation of fractional flow reserve, Sci. Rep. 8 (1) (2018) 1–12.

[10] A.C.I. Malossi, P.J. Blanco, P. Crosetto, S. Deparis, A. Quarteroni, Implicit coupling of one-dimensional and three-dimensional blood flow models with compliant vessels, Multiscale Model. Simul. 11 (2) (2013) 474–506.

[11] S. Sankaran, M.E. Moghadam, A.M. Kahn, E.E. Tseng, J.M. Guccione, A.L. Marsden, Patient-specific multiscale modeling of blood flow for coronary artery bypass graft surgery, Ann. Biomed. Eng. 40 (10) (2012) 2228–2242.

[12] M.E. Moghadam, I.E. Vignon-Clementel, R. Figliola, A.L. Marsden, M. of Congenital Hearts Alliance (MOCHA) Investigators, A modular numerical method for implicit 0D/3D coupling in cardiovascular finite element simulations, J. Comput. Phys. 244 (2013) 63–79.

[13] S. Perotto, A. Ern, A. Veneziani, Hierarchical local model reduction for elliptic problems: a domain decomposition approach, Multiscale Model. Simul. 8 (4) (2010) 1102–1127.

[14] S. Guzzetti, P. Perotto, A. Veneziani, Hierarchical model reduction for incompressible fluids in pipes, Internat. J. Numer. Methods Engrg. 114 (5) (2018) 469–500.

[15] L.A. Mansilla Alvarez, P.J. Blanco, C.A. Bulant, E. Dari, A. Veneziani, R.A. Feijóo, Transversally enriched pipe element method (tepem): An effective numerical approach for blood flow modeling, Int. J. Numer. Methods Biomed. Eng. 33 (4) (2017).

[16] L.A. Mansilla Alvarez, P.J. Blanco, C.A. Bulant, R.A. Feijóo, Towards fast hemodynamic simulations in large-scale circulatory networks, Comput. Methods Appl. Mech. Engrg. 344 (2019) 734–765.

[17] S. Guzzetti, L.A. Mansilla Alvarez, P.J. Blanco, K. Carlberg, A. Veneziani, Propagating uncertainties in large-scale hemodynamics models via network uncertainty quantification and reduced-order modeling, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112626.

[18] Y. Maday, E.M. Rønquist, The reduced basis element method: application to a thermal fin problem, SIAM J. Sci. Comput. 26 (1) (2004) 240–258.

[19] Y. Maday, E.M. Rønquist, A reduced-basis element method, C. R. Math. 335 (2) (2002) 195–200.

[20] L. Iapichino, A. Quarteroni, G. Rozza, A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks, Comput. Methods Appl. Mech. Engrg. 221 (2012) 63–82.

[21] L. Iapichino, G. Rozza, A. Quarteroni, Reduced basis (element) methods for the study of parametrized cardiovascular geometries and networks, 2010.

[22] L. Iapichino, A. Quarteroni, G. Rozza, S. Volkwein, Reduced basis method for the Stokes equations in decomposable domains using greedy optimization, 2014.

[23] E. Løvgren, Y. Maday, E.M. Ronquist, A reduced basis element method for complex flow systems, in: ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond Aan Zee, the Netherlands, September 5-8, 2006, Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.

[24] E. Løvgren, Y. Maday, E.M. Rønquist, A reduced basis element method for the steady Stokes problem: Application to hierarchical flow systems, Model. Identif. Control 27 (2) (2006) 79–94.

[25] S. Deparis, A. Iubatti, L. Pegolotti, Coupling non-conforming discretizations of PDEs by spectral approximation of the Lagrange multiplier space, ESAIM Math. Model. Numer. Anal. 53 (5) (2019) 1667–1694.

[26] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Vol. 92, Springer, 2015.

[27] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer, 2016.

[28] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods, SIAM J. Math. Anal. 43 (3) (2011) 1457–1472.

[29] J.S. Hesthaven, B. Stamm, S. Zhang, Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods, ESAIM Math. Model. Numer. Anal. 48 (1) (2014) 259–283.

[30] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, Numer. Math. 90 (1) (2001) 117–148.

[31] M. Rathinam, L.R. Petzold, A new look at proper orthogonal decomposition, SIAM J. Numer. Anal. 41 (5) (2003) 1893–1925.

[32] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2002) 492–515.

[33] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, R. Scrofani, Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization, J. Comput. Phys. 315 (2016) 609–628.

[34] M. Pfaller, M. Cruz Varona, J. Lang, C. Bertoglio, W.A. Wall, Using parametric model order reduction for inverse analysis of large nonlinear cardiac simulations, Int. J. Numer. Methods Biomed. Eng. 36 (4) (2020) e3320.

[35] G.H. Golub, C.F. Van Loan, Matrix Computations, Vol. 3, JHU press, 2012.

[36] L.N. Trefethen, D. Bau III, Numerical Linear Algebra, Vol. 50, Siam, 1997.

[37] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. Acad. Sci. Paris 339 (9) (2004) 667–672.

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

[39] F. Negri, A. Manzoni, D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, J. Comput. Phys. 303 (2015) 431–454.

[40] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Vol. 44, Springer, 2013.

[41] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers, Rev. Fr. Autom. Inform. Rech. Opér. Anal. Numér. 8 (R2) (1974) 129–151.

[42] P. Hood, C. Taylor, Navier–Stokes equations using mixed interpolation, Finite Elem. Methods Flow Probl. (1974) 121–132.

[43] B. Wohlmuth, A mortar finite element method using dual spaces for the Lagrange multiplier, SIAM J. Numer. Anal. 38 (3) (2000) 989–1012.

[44] C. Bernardi, A new nonconforming approach to domain decomposition: the mortar element method, Nonlinear Partial Equations Appl. (1989).

[45] C. Bernardi, Y. Maday, F. Rapetti, Basics and some applications of the mortar element method, GAMM-Mitt. 28 (2) (2005) 97–123.

[46] D. Braess, W. Dahmen, C. Wieners, A multigrid algorithm for the mortar finite element method, SIAM J. Numer. Anal. 37 (1) (1999) 48–69.

[47] S. Deparis, D. Forti, P. Gervasio, A. Quarteroni, INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces, Comput. & Fluids 141 (2016) 22–41.

[48] P. Gervasio, A. Quarteroni, Analysis of the Internodes Method for Non-Conforming Discretizations of Elliptic Equations, MATHICSE report, 2016.

[49] I. Babuška, The finite element method with lagrangian multipliers, Numer. Math. 20 (3) (1973) 179–192.

[50] T. Zhu, S.N. Atluri, A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free galerkin method, Comput. Mech. 21 (3) (1998) 211–222.

[51] I. Babuška, The finite element method with penalty, Math. Comput. 27 (122) (1973) 221–228.

[52] C.F. Dunkl, Y. Xu, Orthogonal Polynomials of Several Variables, Cambridge University Press, 2014, Number 155.

[53] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.

[54] A. Segal, M. Rehman, C. Vuik, Preconditioners for incompressible Navier–Stokes solvers, Numer. Math.: Theory Methods Appl. 3 (3) (2010) 245–275.

[55] J. Liu, W. Yang, M. Dong, A.L. Marsden, The nested block preconditioning technique for the incompressible Navier–Stokes equations with emphasis on hemodynamic simulations, 2019, arXiv preprint arXiv:1911.10814.

[56] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469.

[57] K. Carlberg, M. Barone, H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, J. Comput. Phys. 330 (2017) 693–734.

[58] N. Dal Santo, S. Deparis, A. Manzoni, A. Quarteroni, An algebraic least squares reduced basis method for the solution of nonaffinely parametrized stokes equations, Comput. Methods Appl. Mech. Engrg. 344 (2019) 186–208.

[59] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1136–1161.

[60] N. Dal Santo, A. Manzoni, Hyper-reduced order models for parametrized unsteady Navier–Stokes equations on domains with variable shape, 2018.

[61] G. Rozza, On optimization, control and shape design of an arterial bypass, Internat. J. Numer. Methods Fluids 47 (10–11) (2005) 1411–1419.

[62] L. Bertagna, S. Deparis, L. Formaggia, D. Forti, A. Veneziani, The LifeV library: engineering mathematics beyond the proof of concept, 2017, arXiv preprint arXiv:1710.06596.

[63] A. Updegrove, N.M. Wilson, J. Merkow, H. Lan, A.L. Marsden, S.C. Shadden, Simvascular: an open source pipeline for cardiovascular simulation, Ann. Biomed. Eng. 45 (3) (2017) 525–541.

[64] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (1–4) (2007) 173–201.