ORIGINAL PAPER

Reduced-order modeling of blood flow for noninvasive functional evaluation of coronary artery disease

Stefano Buoso[1,2] · Andrea Manzoni[3] · Hatem Alkadhi[2] · André Plass[4] · Alfio Quarteroni[3] · Vartan Kurtcuoglu[1,5,6]

Abstract

We present a novel computational approach, based on a parametrized reduced-order model, for accelerating the calculation of pressure drop along blood vessels. Vessel lumina are defined by a geometric parametrization using the discrete empirical interpolation method on control points located on the surface of the vessel. Hemodynamics are then computed using a reduced-order representation of the parametrized three-dimensional unsteady Navier–Stokes and continuity equations. The reduced-order model is based on an *offline–online* splitting of the solution process, and on the projection of a finite volume full-order model on a low-dimensionality subspace generated by proper orthogonal decomposition of pressure and velocity fields. The algebraic operators of the hemodynamic equations are assembled efficiently during the *online* phase using the discrete empirical interpolation method. Our results show that with this approach calculations can be sped up by a factor of about 25 compared to the conventional full-order model, while maintaining prediction errors within the uncertainty limits of invasive clinical measurement of pressure drop. This is of importance for a clinically viable implementation of noninvasive, medical imaging-based computation of fractional flow reserve.

✉ Stefano Buoso
stefano.buoso@uzh.ch

✉ Vartan Kurtcuoglu
vartan.kurtcuoglu@uzh.ch

1 The Interface Group, Institute of Physiology, University of Zurich, Zurich, Switzerland

2 Institute of Diagnostic and Interventional Radiology, University Hospital Zurich, Zurich, Switzerland

3 Chair of Modeling and Scientific Computing, Mathematics Institute of Computational Science and Engineering, École Fédérale Polytechnique de Lausanne, Lausanne, Switzerland

4 Clinic for Cardiovascular Surgery, University Hospital Zurich, Zurich, Switzerland

5 National Center of Competence in Research, Kidney.CH, Zurich, Switzerland

6 Zurich Center for Integrative Human Physiology, University of Zurich, Zurich, Switzerland

1 Introduction

Coronary artery disease (CAD), often manifested by arterial stenosis, is one of the leading causes of mortality worldwide (Heidenreich et al. 2011). Coronary artery stenosis can result in reduced blood flow to the region of the myocardium supplied by the narrowed vessel branch. Clinically, the functional severity of the stenosis can be quantified with the fractional flow reserve (FFR) index. FFR is calculated as the ratio between the blood pressure distal to a stenosis and the aortic blood pressure, both of which are invasively measured under hyperaemic conditions using a pressure wire catheter (Gould et al. 1974). Several clinical studies have shown the effectiveness of such functional evaluation in driving lesion treatment with improved clinical outcome compared to standard angiographic assessment (Pijls et al. 2007; Tonino et al. 2010; Heidenreich et al. 2011; Zimmermann et al. 2015). However, using a catheter can overestimate lesion severity by causing an obstruction that increases the pressure drop by up to 20% depending on the vessel

anatomy and blood flow conditions (Ashtekar et al. 2007; de Vecchi et al. 2014).

To address the invasiveness of the procedures, fractional flow reserve estimation based on computational hemodynamic models (FFRc) has been proposed and clinically validated (Taylor et al. 2013; Coenen et al. 2015; Hlatky et al. 2015; Douglas et al. 2016; Zhang et al. 2016). Yet, determining FFRc is computationally expensive, requiring either high performance computing resources or long times to solution, and neither is compatible with clinical routine for the assessment of CAD (Zhang et al. 2016). Aiming at reducing the computational burden of these calculations, Itu et al. (2016) have proposed a data-driven modeling approach based on a neural network for the prediction of FFRc without performing full hemodynamic simulations. While this approach allows for almost *real-time* assessment of stenosis severity, the FFR estimate is based on restrictive assumptions on the shape of the vessel and on a simplified one-dimensional hemodynamic model. In this work, we propose a different approach: instead of a surrogate physics representation derived from training a neural network, we employ a model to compute the full three-dimensional hemodynamic flow field in coronary arteries, starting directly from the mass and momentum conservation equations. Another advantage of this approach compared to those based on machine learning is that it provides the flow field in the full domain, allowing to compute various metrics a posteriori without having to train a new algorithm. In addition, we introduce a flexible, general parametric description of the computational domain and boundary conditions to represent anatomic and physiologic features of blood vessels.

The issue of high computational cost and long time to solution is a recurring problem for systems described by partial differential equations (PDEs) in general. When the systems further depend on adjustable parameters, as in the case of the identification of optimal simulation parameters (Olgac et al. 2009; Knight et al. 2010; Rikhtegar et al. 2012) and subject-specific evaluations and surgical planning (Gijsen et al. 2014; Vergallo et al. 2014; de Zélicourt and Kurtcuoglu 2016), evaluating the full-order model (FOM) described by the discretized PDEs can become prohibitively expensive. To address this problem, reduced basis (RB) methods for parametrized partial differential equations and, more generally, reduced-order models (ROMs), have been studied intensively in the last decade, e.g., to enable faster calculations of fluid dynamics problems (Quarteroni and Rozza 2007; Rowley 2011; Bergmann et al. 2009; Amsallem et al. 2012; Manzoni 2014; Ballarin et al. 2015; Buoso and Palacios 2017). While attempts have been made to use ROMs for subject-specific computational fluid dynamics (CFD) (Manzoni et al. 2012a, b; Lassila et al. 2013; Colciago et al. 2014; Ballarin et al. 2016, 2017), these methodologies have shown limited capability in aiding clinicians

in the subject-specific assessment of stenosis severity. The challenge, in fact, arises from the need to represent often quite different subject-specific geometries and data under a common framework. So far, this point has required (1) heavy computational preprocessing to describe patients anatomic and physiologic variability in terms of meaningful parameters, and (2) cumbersome operations to adapt the resulting model to patient-specific cases during the solution process.

Here, we propose to use a reduced-order hemodynamic model to predict the pressure drop along stenosed vessels. The hemodynamic equations are discretized with the finite volume (FV) method and projected into a subspace of lower dimension, enabling fast calculations. Previous RB hemodynamic models are almost entirely based on the finite element method (Manzoni et al. 2012a, b; Lassila et al. 2013; Colciago et al. 2014; Ballarin et al. 2016, 2017). Existing RB approaches based on the FV method (Stabile et al. 2015; Stabile and Rozza 2018) do not allow to define arbitrary three-dimensional geometric parametrizations, thus limiting their adoption for patient-specific cases. Moreover, they address the pressure–velocity coupling using a pressure equation different from the one solved in the full-order FV model. In contrast, our computational pipeline builds on the exact algebraic form of the unsteady incompressible Navier–Stokes and continuity equations (hereinafter collectively abbreviated as NS) used in the full-order model (FOM), wherefrom we build a nonlinear RB-ROM based on *offline–online* splitting of the computations. System dimensionality is reduced by projecting the FOM onto a subspace obtained from proper orthogonal decomposition (POD) of a dataset of velocity and pressure fields. The algebraic operators of the equations in the ROM are then assembled efficiently, i.e., with low computational cost, in the *online* phase using the discrete empirical interpolation method (DEIM) (Maday et al. 2009). DEIM is also employed, for the first time, to provide a low-dimensional description of the computational domain which can integrate arbitrary parametrizations with the reduced-order description of the fluid-dynamic equations. The aim of our new method is the fast, subject-specific computation of pressure drop across a stenosis.

The manuscript is structured as follows: Sect. 2 introduces the conventional method for FFRc calculations and the pipeline for building and using the ROM. We present the results in Sect. 3 and discuss the advantages and limitations of the methodology in Sect. 4.

2 Methods

2.1 Problem description: FFRc calculations

Our approach is based on an *offline–online* splitting of the computations as sketched in the flowchart of Fig. 1. The
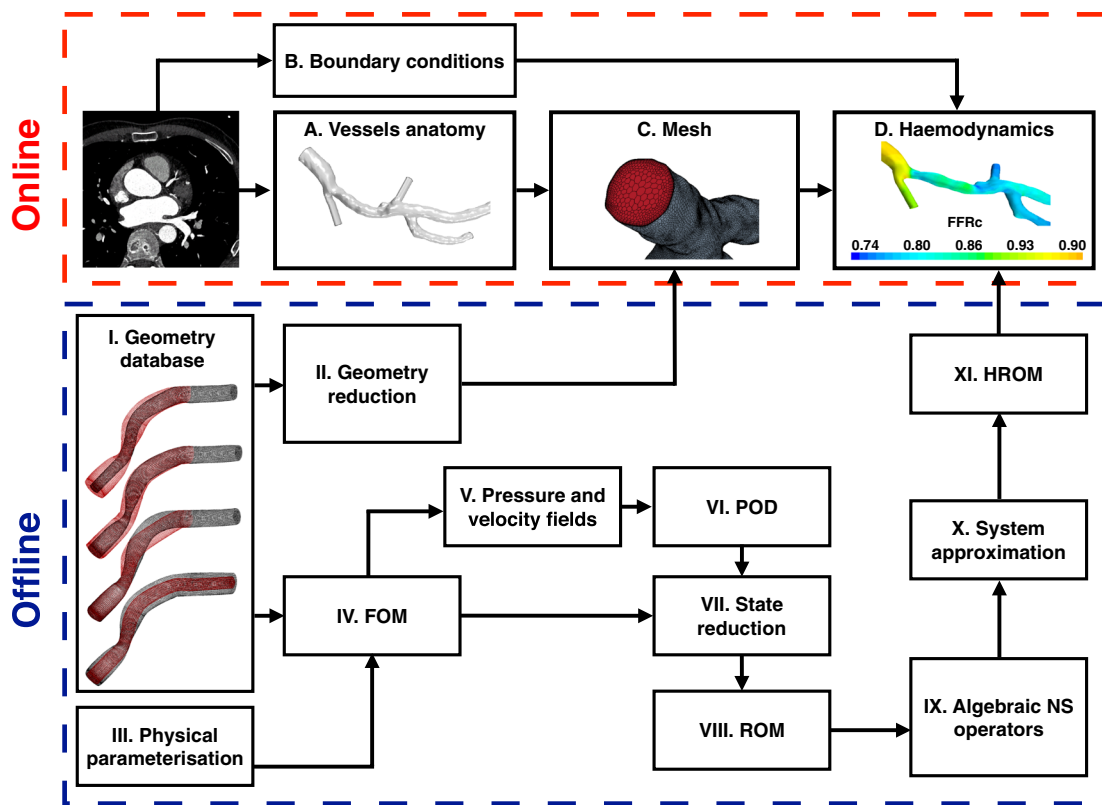
	Step	Description
I.	Geometry database	Generation of geometries with representative anatomical variations.
II.	Geometry reduction	DEIM is used to define a reduced-order geometric description of (I).
III.	Physical parametrization	Definition of a reduced-order description of boundary conditions.
IV.	FOM	Computation of the conventional full-order haemodynamic solution.
V.	Pressure and velocity fields	Storage of pressure and velocity fields from (IV).
VI.	POD	POD to calculate the reduced bases for pressure and velocity fields.
VII.	State reduction	Projection of FOM (IV) onto POD bases (VI) to reduce the dimensionality of the haemodynamic equations (Galerkin projection).
VIII.	ROM	ROM integration to retrieve the solutions from the FOM (IV).
IX.	Algebraic NS operators	Storage of algebraic description of the NS operators from (VIII).
X.	System approximation	DEIM to define a reduced-order description of the NS operators in (IX).
XI.	HROM	Final ROM for use in the *online* phase.

Fig. 1 Pipeline for the calculation of FFRc. The *online* block (steps A to D) corresponds to the conventional calculation process using a full-order model. The *offline* block contains the steps used for the geometric parametrization and the creation of a reduced-order hemo-dynamic model. The two outputs of the *offline* block are the reduced-order geometric description from the geometry reduction (II) and the HROM (XI), which are used in *online* steps C and D, respectively, for the calculation of subject-specific FFRc

online block includes the standard steps for the calculation of the subject-specific FFR. The process starts with the acquisition of images (e.g., using computed tomography) and the segmentation of the branches of interest (A). Secondly, a computational domain is created from the vessel lumen (C) and used for the discretization of the incompressible NS equations according to the selected numerical strategy. We will refer to this model as the full-order model (FOM). The numerical procedure requires the coupling of the resulting system of equations with boundary conditions, which are derived from subject-specific noninvasive clinical images and measurements (B). To this end, usually, the inlet and outlet boundaries of the computational domain are coupled with lumped-parameter

models of the heart and coronary microcirculation, respectively (Stergiopulos et al. 1995; Sankaran et al. 2012; Taylor et al. 2013). The NS equations are then solved and the FFRc is computed (D).

Our aim is to speed up steps C and D of Fig. 1 by using the aforementioned *offline–online* decomposition strategy, where the *offline* phase, performed only once during the development of the model, accounts for the highest computational burden, but leads to faster *online* subject-specific hemodynamic calculations. Our *offline* stage starts from a dataset of computational domains sharing the same node connectivity and describing the variability of the vessels of interest (I). We then apply DEIM to the vertex coordinates of the domains in (I) to obtain a set of geometrical bases that, weighted by appropriate coefficients, can reconstruct the domains of interest up to a preselected approximation tolerance (II). Such reduced-order geometric description can efficiently reconstruct the computational domain inside a given vessel reducing the computational time of the step (C) in the *online* phase. In the *offline* step (III), we parametrize the boundary conditions for the blood vessels which, in general, will depend on the specific application considered. In our application, we use the mean blood flow velocity as unique parameter.

The geometric and physical parameters identified in steps (I) and (III), respectively, are used to create a predefined set of FOMs of the population for which we compute the corresponding hemodynamics using a FV solution strategy (IV) and store the corresponding velocity and pressure fields (V). Based on these solutions (or *snapshots*), we use POD to generate a set of bases (VI) whose linear combination can reconstruct the original solutions. By imposing orthogonality of the algebraic formulation of the FOM residuals on the POD bases (Galerkin projection), we can reduce the system's dimensionality of states (VII). This leads to a reduced-order model with low dimensionality, which we refer to as ROM (VIII). We integrate the ROM for the cases considered in (IV) and store the algebraic NS operators (IX). Using DEIM on these matrices (X), we determine an approximation of the algebraic description of the NS operators, which will allow their efficient assembly during the *online* solution process (XI). The ROM using the DEIM approximation to reconstruct the operators is referred to as hyper-reduced ROM (HROM). With this final ROM, the time to solution of subject-specific calculations will be reduced.

In the remainder of this section, we will detail the steps presented in Fig. 1. We remark that in order to calculate FFR, both the pressure drop along the vessel and the aortic pressure must be known. Here, we will focus on the computational pipeline for the prediction of the pressure drop. We have developed the pipeline based on simplified yet

realistic coronary artery branch anatomies, and tested it for large pressure drops as they may occur in severe stenoses.

2.2 Geometry database

We populated the geometric database (Fig. 1, I) with synthetic geometries approximating the variability of the most important anatomic features of coronary arteries, such as diameter scaling and tapering, stenosis severity, position and length. The geometric parametrization is set a priori during the generation of the database. However, this information is not provided explicitly to our geometric parametrization algorithm. Instead, the algorithm automatically identifies a set of parameters and suitable mapping from those to the mesh point coordinates based on the used geometries. This allows to easily adapt the parametrization to general shapes for which a parametric description is not known or not imposed a priori.

All synthetic geometries in our study are obtained from the deformation of a three-dimensional straight pipe with diameter $d_0 = 4$ mm and length $L_0 = 40$ mm, which is shown schematically in Fig. 2. A Cartesian coordinate system is defined with origin at the center of the inlet section and the x_3 axis along the length of the pipe. The fluid domain is meshed with 75,000 hexahedral elements, which have been shown to provide grid-independent pressure drop results for the geometries and fluid conditions addressed in this work. The mesh is generated using the native OpenFOAM structured mesher (Weller et al. 1998).

Geometric features are then added to the reference mesh to represent realistic branch anatomies. First, diameter scaling and a bell-shaped section restriction along the x_3 axis are prescribed with the mapping function

$$x_i = x_i \mu_{g,0} \left[1 - \mu_{g,1} \exp\left(-\frac{(x_3 - \mu_{g,3} L_0)^2}{\mu_{g,2} L_0} \right) \right] \qquad \text{and} \qquad i = 1, 2,$$

(1)

where x_i are the coordinates of the mesh points, $\mu_{g,0}$ is the diameter scaling factor and $\mu_{g,1}$, $\mu_{g,2}$ and $\mu_{g,3}$ are the
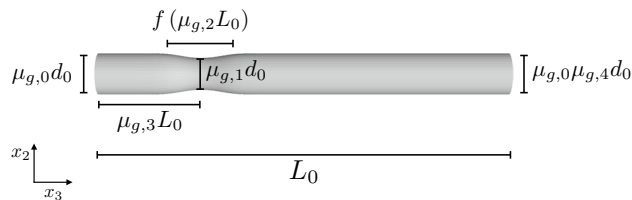

Fig. 2 Reference domain (representing a coronary artery branch) with added features (changes of the geometry due to stenosis and tapering): $\mu_{g,0} d_0$ is the inlet diameter, $\mu_{g,0} \mu_{g,4} d_0$ the outlet diameter, $\mu_{g,1} d_0$ the diameter of the stenosis throat, $\mu_{g,3} d_0$ the throat position on the x_3 axis, L_0 is the total length (constant for all geometries). The stenosis length is a function of $\mu_{g,2} L_0$

amplitude, length and throat position parameters of the bell-shaped restriction, respectively. In what follows, we will use $\mu_{g,1}$ to represent the stenosis severity based on the diameter. Secondly, tapering of the channel is prescribed with a linear function defined between the throat position, $\mu_{g,3}L_0$, and the domain outlet such that

$$x_i = x_i \mu_{g,4} \left[\frac{z - \mu_{g,3}L_0}{L_0\left(1 - \mu_{g,3}\right)} \right] \qquad \text{with} \qquad z \geq \mu_{g,3}L_0, \qquad i = 1, 2,$$
(2)

where $\mu_{g,4}$ is to the ratio between the outlet diameter and the inlet one.

Each element of the geometric database, $X_i = X\left(\mu_g^i\right) \in \mathbb{R}^{h \times 3}$ is obtained applying the transformations of Eqs. (1) and (2) with a prescribed set of the aforementioned parameters, μ_g^i. Specifically, $X_i(j, :) = \left(x^j, y^j, z^j\right)$ are the new coordinates of the jth point of the reference computational domain after the transformation.

2.3 Physical parametrization

We impose physiologic blood flow conditions in the vessels by prescribing the inlet velocity profile. For our simulations, we have set a constant parabolic inlet velocity profile since this has been seen to be a reasonable assumption for FFR calculations (Zhang et al. 2016): clinically, FFR is obtained as the time-average of the values measured during the diastolic phase of the cardiac cycle, when coronary flow and microcirculation resistance are almost constant. Based on these assumptions and considering the cylindrical pipe configuration introduced in Sect. 2.2, the inlet velocity $\boldsymbol{u}_{\text{in}} = \left(u_{1_{\text{in}}}, u_{2_{\text{in}}}, u_{3_{\text{in}}}\right)$ has a nonzero component in the x_3 direction and is defined as

$$u_{3_{\text{in}}}\left(\mu_g^i, \mu_p^i\right) = 2\mu_p \left(1 - 4 \frac{\left(x_1^2\left(\mu_g^i\right) + x_2^2\left(\mu_g^i\right)\right)}{d^2\left(\mu_g^i\right)} \right)\Bigg|_{x_3\left(\mu_g^i\right)=0}$$
(3)

where $x_1\left(\mu_g^i\right)$, $x_2\left(\mu_g^i\right)$ and $x_3\left(\mu_g^i\right)$ are the coordinates of a point of the domain, $d\left(\mu_g^i\right)$ is the inlet diameter and the physical parameter μ_p is the mean inlet velocity. As highlighted in the notation, geometry and point coordinates depend on the geometric parameters describing the configuration of the selected domain. While we have considered here an axis-symmetric parabolic profile, more complex inlet flow profiles could be employed just as well.

2.4 Full-order hemodynamic model and finite-volume solution

After the definition of a domain $\Omega\left(\mu_g\right)$ with boundaries $\partial\Omega\left(\mu_g\right)$ described by the selected geometric parametrization, we set the unsteady, incompressible parametrized NS and continuity equations to model the blood flow in the vessel (required for the *online* step C and *offline* step IV in Fig. 1). Blood is assumed to behave like a Newtonian incompressible fluid with constant and uniform density, ρ, and kinematic viscosity, ν. The system of equations, under rigid wall assumptions, reads as follows

$$\begin{cases} \frac{\partial}{\partial t}\boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{0} & \text{in } \Omega\left(\mu_g\right) \times (0, T) \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega\left(\mu_g\right) \times (0, T), \\ \boldsymbol{u} = \boldsymbol{u}_{\text{in}}\left(\mu_p\right) & \text{in } \partial\Omega_D\left(\mu_g\right) \times (0, T), \\ -p\boldsymbol{n} + \nu(\nabla\boldsymbol{u})\boldsymbol{n} = \boldsymbol{h}\left(\mu_p\right) & \text{in } \partial\Omega_N\left(\mu_g\right) \times (0, T), \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{in } \Omega\left(\mu_g\right) \times t = 0, \end{cases}$$
(4)

where $\boldsymbol{u}$ is the fluid velocity, p is the fluid pressure divided by the fluid density, $\boldsymbol{n}$ is the boundary normal vector, $\boldsymbol{u}_{\text{in}}\left(\mu_p\right)$ is the boundary condition imposed at inlet and vessel walls ($\partial\Omega_D$) and $\boldsymbol{h}\left(\mu_p\right)$ is the boundary condition at the outlet ($\partial\Omega_N$), which we set to zero in our work. For clarity, we have explicitly specified the dependence of the computational domain on the geometrical parameters, μ_g, and of the boundary conditions on physical parameters, μ_p. Hereinafter, unless specifically stated, μ will refer to the generic dependence on the parameters.

The pressure–velocity coupling is obtained using the PISO algorithm, in which mass conservation is imposed by solving a pressure-Poisson equation (Jasak 1996; Moukalled et al. 2015). This method allows resolving the saddle-point problem of the system of equations and couple the solution of the pressure and velocity fields. The algebraic descriptions of these operators can be derived after the selection of suitable spatial and temporal discretization. Here, linear interpolation and central differencing are used for spatial discretization. A semi-implicit time-marching scheme is used where solution-dependent terms need to be linearized with respect to the solution of the last time step. After spatial and time discretization, the algebraic form of the incompressible NS equations yielding our FOM reads as

$$\begin{bmatrix} M_h + C_h\left(\boldsymbol{u}_h^t\right) - \nu\boldsymbol{\Delta}_h & \boldsymbol{\nabla}_h \\ \boldsymbol{0} & \boldsymbol{\Delta}_h\left(\boldsymbol{u}_h^t\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h^{t+\Delta t} \\ p_h^{t+\Delta t} \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{u},h}^{t+\Delta t} \\ f_{p,h}^{t+\Delta t} \end{bmatrix}^{BC_D} + \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{u},h}^t \\ f_{p,h}^t \end{bmatrix}^{BC_N},$$
(5)

where $[\bullet]^t$ and $[\bullet]^{t+\Delta t}$ refer to solution fields at the previous and current time steps, respectively, $\boldsymbol{u}_h \in \mathbb{R}^{3h}$ and $p_h \in \mathbb{R}^h$

are the velocity and pressure fields at the mesh cell centroids, $\boldsymbol{M}_h$, $\boldsymbol{C}_h(\boldsymbol{u}_h)$, $\nu\boldsymbol{\Delta}_h$ and $\boldsymbol{\nabla}_h$ are the mass, convective, Laplacian and pressure-gradient operators of the momentum conservation equation and $\boldsymbol{\Delta}_h(\boldsymbol{u}_h)$ the modified Laplacian of the pressure-Poisson equation. In the notation used, the dependence on the time increment Δt is implicitly included in the mass matrix. The two terms at the right-hand side represent the discretized versions of the Dirichlet ($[\bullet]^{BC_D}$) and Neumann ($[\bullet]^{BC_N}$) boundary conditions, respectively. At each time step, the system is linearized using the previous solution for the evaluation of $\boldsymbol{C}_h(\boldsymbol{u}_h)$, $\boldsymbol{\Delta}_h(\boldsymbol{u}_h)$ and $[\bullet]^{BC_N}$. The subscript h refers to the discretization of the equations in the FOM computational domain, h being the total number of cells of the domain. We have used the OpenFOAM FV framework to discretize and solve the NS equations (Weller et al. 1998). For a detailed derivation of the equations, the form of the discretized operators and the PISO solution process, the reader is referred to (Jasak 1996; Moukalled et al. 2015). Since our aim is to compute the pressure drop along the vessel, we use herein a zero-pressure reference at the domain outlet. However, any reference pressure could be used, including one obtained by coupling the stenosis model with a lumped-parameter representation of coronary circulation as shown in Stergiopulos et al. (1995), Sankaran et al. (2012) and Taylor et al. (2013).

2.5 Dimensionality reduction

This section introduces the POD and DEIM methods used during the *offline* phase to reduce the dimensionality of both geometric and hemodynamic descriptions. Concretely, POD is used to construct the reduced bases for the velocity and the pressure fields through the so-called *method of snapshots* (Buljak 2011) (Fig. 1, VI). These subspaces are used for the Galerkin projection (Fig. 1, VII) of the algebraic NS equations to reduce the state of the system (Fig. 1, VIII). Additionally, the application of POD to the elements of the geometric dataset, $X_n = X\left(\boldsymbol{\mu}_g^n\right)$, or to the discrete fluid-dynamic operators of the NS equations, will generate the bases which, together with the DEIM coefficients (Fig. 1, steps II, X), will allow the definition of an eicient sampling strategy for their *online* reconstruction.

2.5.1 Proper orthogonal decomposition

To ensure that this article is self-contained, we review here, briefly, the construction of a reduced-order approximation of a generic parametric-dependent field through POD. In our case this field, $\mathbf{z}(\boldsymbol{\tau})$, could consist of hemodynamic velocities, $\boldsymbol{u}_h$, pressures, p_h, mesh point coordinates, X_n, or discrete operators from Eq. (5). Let $\mathbf{z}(\boldsymbol{\tau})$ be a map defined

on $\mathcal{D}$ with values in $\mathbb{R}^h$ and let $\mathcal{M}_z = \{\mathbf{z}(\boldsymbol{\tau}) \in \mathbb{R}^h \mid \boldsymbol{\tau} \in \mathcal{D}\}$ be the manifold identified by the image of $\mathbf{z}$. The goal of POD is to approximate $\mathcal{M}_z$ with a low-dimensional optimal linear space of $\mathbb{R}^h$ starting from a set of n_s *snapshots* $\mathbf{z}_i = \mathbf{z}(\boldsymbol{\tau}_i)$, with $\boldsymbol{\tau}_i \in \mathcal{D}$, $i = 1, \ldots, n_s$. Here, $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{n_s}$ are randomly sampled points of $\mathcal{D}$; other strategies, such as *latin hypercube sampling* or *sparse grid* techniques, could be exploited. These strategies become especially useful for high-dimensional parameter spaces, where the system might exhibit a non-smooth response, or where it may have different physical regimes in subregions of the parameter space (Pagani et al. 2018). Using the *snapshots* method (Buljak 2011), the POD bases are computed from the eigenvectors, $\boldsymbol{\xi}_i$ and eigenvalues, σ_i of the correlation matrix $\mathbf{C} = \mathbf{Z}^T \mathbb{X}_h \mathbf{Z} \in \mathbb{R}^{n_s \times n_s}$, where $\mathbf{Z}$ is the snapshot matrix

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{n_s}] \in \mathbb{R}^{h \times n_s},$$

and $\mathbb{X}_h \in \mathbb{R}^{h \times h}$ is a symmetric positive-definite matrix encoding a suitable norm. Since $\mathbf{C}$ is symmetric and positive-definite, we have that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_s} \geq 0$ and the POD bases are obtained as

$$\boldsymbol{\phi}_i = \frac{1}{\sqrt{\sigma_i}} \mathbf{Z} \boldsymbol{\xi}_i, \quad i = 1, \ldots, N \leq N_h$$

from the N eigenvectors $\boldsymbol{\xi}_i$ corresponding to the first N (largest) singular values; we can set the bases dimension N as the minimum integer such that

$$\frac{\sum_{i=1}^N \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} \geq 1 - \varepsilon_{\text{POD}} \tag{6}$$

for a given small tolerance $\varepsilon_{\text{POD}} > 0$. For time-independent field variables and operators, $\boldsymbol{\tau} \equiv \boldsymbol{\mu}$, POD is performed with the procedure described above. For time-dependent field variables, as in the case of velocity, pressure, convective and pressure-Laplacian algebraic NS operators and discretised Neumann boundary conditions, $\boldsymbol{\tau} \equiv [t, \boldsymbol{\mu}]$ and the procedure relies on snapshots taken of each different parameter value combination at different instances in time. Then, the correlation matrix C is assembled either once, on the global matrix collecting all the snapshots (corresponding to $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{\text{ns}}$ at times $t^1, \ldots, t^{N_t}$), or sequentially, taking for each selected parameter values the snapshots corresponding to different instants, and finally performing POD again of the retained eigenmodes for each $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{\text{ns}}$. Here, we have selected the first approach and computed the POD bases on the global correlation matrix. The second approach is preferable when there is insufficient memory to store all snapshots at once.

In the spirit of POD decomposition, each (possibly time-dependent) instance in $\mathcal{M}_z$ can be approximated as

$$\mathbf{z}(\boldsymbol{\tau}) \approx \mathbf{z}_r(\boldsymbol{\tau}) = \sum_{i=1}^{N} \boldsymbol{\phi}_i a_i(\boldsymbol{\tau}) = \boldsymbol{\Phi}\mathbf{a}(\boldsymbol{\tau}), \tag{7}$$

where $\boldsymbol{\Phi}_N = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N] \in \mathbb{R}^{h \times N}$ are parameter and time-independent bases and $\mathbf{a}(\boldsymbol{\tau})$ the vector whose components are the respective amplitudes.

2.5.2 Galerkin projection

The aim of the Galerkin projection (Fig. 1, VII) is to reduce the number of FOM states by imposing normality of the algebraic formulation of the FOM residuals with respect to a suitable bases subspace. In our case, the bases are generated using POD on snapshots of velocity and pressure fields (output of step VI in Fig. 1) obtained from the solutions of the full-order model (FOM) for selected sampled combinations of the physical and geometrical parameters. The corresponding projection matrices from the full to the reduced-order spaces are $\boldsymbol{\Phi}_u = \begin{bmatrix} \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N_v} \end{bmatrix} \in \mathbb{R}^{h \times N_v}$ and $\boldsymbol{\Phi}_p = \begin{bmatrix} \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N_p} \end{bmatrix} \in \mathbb{R}^{h \times N_p}$ with $N_v \neq N_p$ a priori.

The Galerkin projection of the FOM of Eq. (5) onto the RB spanned by the columns $\boldsymbol{\Phi}_u$ and $\boldsymbol{\Phi}_p$ yields, algebraically, the following ROM

$$\begin{bmatrix} M_N + C_N(u_h^t) - \nu\Delta_N & \nabla_N \\ 0 & \Delta_N(u_h^t) \end{bmatrix} \begin{bmatrix} u_N^{t+\Delta t} \\ p_N^{t+\Delta t} \end{bmatrix} \\ = \begin{bmatrix} f_{u,N}^{t+\Delta t} \\ f_{p,N}^{t+\Delta t} \end{bmatrix}^{BC_D} + \begin{bmatrix} f_{u,N}^{t} \\ f_{p,N}^{t} \end{bmatrix}^{BC_N}, \tag{8}$$

where the same notation as in Eq. (5) has been adopted. M_N, $C_N(u_h)$, $\nu\Delta_N$ and ∇_N are the reduced-order representation of the mass, convective, Laplacian and pressure-gradient operators of the momentum conservation equation and $\Delta_N(u_h)$ the reduced-order representation of the modified Laplacian of the pressure-Poisson equation. The two terms at the right-hand side represent the projection of discretized versions of the Dirichlet ($[\bullet]^{BC_D}$) and Neumann ($[\bullet]^{BC_N}$) boundary conditions onto the corresponding RB. The subscript h and N refer to the number of degrees of freedom of the FOM and ROM, respectively.

The dimension of the system defined by Eq. (8) is now much lower than that of the FOM (Eq. (5)), and solving the system has become computationally less expensive. However, without further development, the ROM still requires the calculation of all operators in the FOM ahead of their projection onto the POD bases. Additionally, some of the operators in Eq. (5) depend on the FOM solution (dependence on u_h in Eq. (8)) so that they would need to be recomputed at each iteration using the FOM and then projected onto the POD subspace. This would translate to only a marginal cost reduction

when using the ROM as in Eq. (8). For this reason, we need a method to efficiently reconstruct these solution-dependent operators at each iteration. We refer the reader to the work of Negri et al. (2015) for more detail.

2.5.3 Discrete empirical interpolation method

During the *online* phase, the generation of both computational domain and algebraic operators of the NS equations must be performed very efficiently. We use DEIM (Maday et al. 2009) (or MDEIM, for its matrix version) to reduce the computational cost of these two tasks and obtain the approximate mesh coordinates and NS operators. We will generally refer to them as $\mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$, where $\mathbf{u}_h(\boldsymbol{\mu})$ specifies the dependence on the solution field and $\boldsymbol{\mu}$ the dependence on the parameters describing the problem. We seek to approximate $\mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$ as a linear combination of $(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$-independent terms, $\boldsymbol{\phi}_R^i$, and corresponding $(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$-dependent weights, $\theta_R^i(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$,

$$\begin{aligned} \mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) \approx \mathbf{A}_r(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) &= \boldsymbol{\Phi}_R\theta_R(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) \\ &= \sum_{i=1}^{m_R} \theta_R^i(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})\boldsymbol{\phi}_R^i, \end{aligned} \tag{9}$$

where $\theta_R(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{m_R}$ is the vector of the coefficients to be determined and $\boldsymbol{\Phi}_R = \begin{bmatrix} \boldsymbol{\phi}_R^0, \ldots, \boldsymbol{\phi}_R^{m_r} \end{bmatrix}$.

The bases $\boldsymbol{\Phi}_R$ can be computed during the *offline* phase by performing POD on a set of snapshots of $\mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$ as described in Sect. 2.5.1. The coefficient vector $\theta_R(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$ can be evaluated for each new value combination of $\boldsymbol{\mu}$ by imposing m_R interpolation constraints on a subset $\wp = [\wp_1, \ldots, \wp_{m_R}]$ of entries of $\mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})$ [the so-called magic points, see, e.g., Maday et al. (2009)] selected by the DEIM algorithm. Therefore, during the *online* phase, it is sufficient to calculate the algebraic operators only at those points to quickly reconstruct their formulation in the ROM.

2.5.3.1 DEIM for NS algebraic operators To apply DEIM to the matrices corresponding to the algebraic operators in Eq. (5), we calculate the average value, $A_{\boldsymbol{\phi}_R}^0$. Then, we build the snapshot matrix for the POD algorithm (Sect. 2.5.3) by stacking into a column vector the difference between the operator and the mean, i.e., $\mathbf{z}_i = \text{vect}(\mathbf{A}(\mathbf{u}_h(\boldsymbol{\mu}^i);\boldsymbol{\mu}^i) - A_{\boldsymbol{\phi}_R}^0)$. After calculating the bases functions $\boldsymbol{\phi}_R^i$, these can be reassembled into their matrix form, $A_{\boldsymbol{\phi}_R}^i$, and projected onto the relevant POD subspace so that each operator in (8) can be approximated as

$$\begin{aligned} \mathbf{A}_N(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) &\approx \mathbf{A}_{N,r}(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu}) \\ &= \sum_{i=0}^{m_R} \theta_R^i(\mathbf{u}_h(\boldsymbol{\mu});\boldsymbol{\mu})A_{N,\boldsymbol{\phi}_R}^i, \end{aligned} \tag{10}$$

where $A_{N,\phi_s}^i = \Phi_{f_r}^T A_{\phi_R}^i \Phi_{f_l}$ and $\theta_R^0 = 1$. Here, Φ_{f_r} and Φ_{f_l} represent the appropriate subspaces Φ_u or Φ_p, depending on the operator. In this case, magic point subsets correspond to $\wp = [\wp_1, \ldots, \wp_{m_R}]$ element entries of the selected operator. During the *online* phase, modal amplitudes θ_R^i are calculated by imposing that Eq. (9) is satisfied for these selected entries and, finally, the operator is approximated using Eq. (10). The ROM exploiting DEIM approximation of the NS algebraic operators will be referred to as HROM.

2.5.3.2 DEIM for geometric reconstruction We first compute the mean geometry, $\bar{X}_{\mathrm{ref}}$, obtained from the point-wise average of the coordinates of the domains in the dataset. Then, we build the snapshot matrix for the DEIM algorithm (Sect. 2.5.3) by stacking into a column vector the difference between the coordinates of each instance of the geometric database and the mean geometry, i.e., $\mathbf{z}_i = \mathrm{vect}(X(\boldsymbol{\mu}_g^i) - \bar{X}_{\mathrm{ref}})$. The magic points obtained from DEIM identify an equivalent number of vertices of the mesh boundary. As for the NS operators, during the *online* phase, the modal amplitudes θ_R^i are calculated by imposing that Eq. (9) is satisfied for these coordinates of these points. The one-to-one correspondence of the points of the atlas and a general shape will, in general, not be ensured. Therefore, the points used to impose the interpolation constraints will be identified by a suitable mapping of the geometry. In the case of vessels, this could be done by a cylindrical parametrization of atlas and reference geometry. In particular, after the segmentation of the patient anatomy and the isolation of the lesioned branch of interest, the magic points are identified based on the centerline position along the branch and the polar coordinates in a section-based reference frame. We want to highlight that during the *online* phase, we only require the mesh elements necessary for the calculation of the DEIM coefficients of the NS operators. Therefore, to further accelerate the calculations, we can reduce the bases Φ_R to the mesh points defining those cells. In our pipeline the geometric parametrization is used to reconstruct only the reduced meshes for each operator.

3 Results

3.1 Geometry database

We generated 100 synthetic geometries by randomly prescribing the components of the parameter space introduced in Sect. 2.2 within a predefined range. The reference (inlet) diameters vary between 2 and 6 mm, corresponding to $\mu_{g,0} = 1.0 \pm 0.50$, fully covering the variation found in the statistical analysis of coronary arteries by Mancini et al.

Table 1 Parameters and their value ranges for the geometry database. Indicated are also the geometric features they affect

μ_g	Parameter	Range		
		Min	Max	
$\mu_{g,0}$	Inlet diameter	2.0	6.0	mm
$\mu_{g,1}$	Stenosis severity	0.2	0.5	–
$\mu_{g,2}$	Stenosis length	0.5	1.0	10^{-4}
$\mu_{g,3}$	Stenosis position	0.37	0.63	–
$\mu_{g,4}$	Tapering	0.85	1.15	–

(2007). Tapering values reported by the same authors led to the selection of $\mu_{g,4} = 1.0 \pm 0.15$ to represent healthy and diseased physiological tapering conditions. The stenosis severity parameter $\mu_{g,1}$ ranges from 0.2 to 0.5 and the lesion is arbitrarily located along the vessel by selecting $\mu_{g,3}$ values from 0.37 to 0.63. The reference stenosis length parameters, $\mu_{g,2}$, is varied in the range [0.5–1.0]×10^{-4}. The values are summarized in Table 1.

3.2 Geometric parametrization and reconstruction

The geometric parametrization is obtained by applying DEIM to 75 out of the 100 geometries generated in the database, while the 25 excluded cases are used to assess the performance of the parametrization on new geometries. Setting a tolerance of 10^{-2}, 10^{-3} and 10^{-4} for DEIM leads to the selection of $K = 3$, 5, and 7 bases, respectively. The mean geometry, $\bar{X}_{\mathrm{ref}}$, is shown in Fig. 3a where we have also highlighted the position of the resulting magic points for a representative case with $\varepsilon = 10^{-4}$, which we further use to illustrate our results. In Fig. 3, we present the individual contribution of the first two modes, $\boldsymbol{\phi}_{R\,i=1\ldots2}^i$, over the mean shape for the maximum and minimum values of the corresponding coefficient θ_R^i obtained from DEIM.

Figure 4 illustrates the error in the reconstruction on the training geometries (left panel) and on a test-set (right) for the three different tolerances. Errors are computed as the maximum distance between two corresponding nodes and shown relative to the reference diameter, d_0.

3.3 ROM training

For each of the defined training geometries, we have set a constant parabolic inlet velocity profile with a mean velocity magnitude between 0.2 and 1.2 ms^{-1}, which corresponds to flow regimes with Reynolds numbers (based on the reference inlet diameter d_0) between 200 and 1200. The lower end of this range corresponds to resting physiologic conditions in coronary arteries as derived from the coronary blood flow measurements from Keegan et al. (2004) and Zafar et al. (2014). The maximum velocity for the training of

Fig. 3 Output of geometric parametrization: **a** mean shape and magic points (red spheres) for the case with $\varepsilon = 1 \times 10^{-4}$, **b** first and **c** second reconstruction modes: the yellow shape corresponds to the minimum value of the reconstruction coefficient, the blue shape to the maximum value
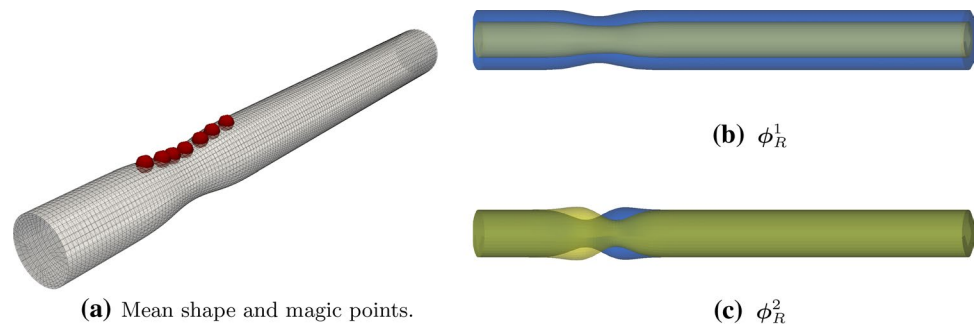

(a) Mean shape and magic points.

(b) ϕ_R^1

(c) ϕ_R^2

Fig. 4 Reconstruction errors for **a** training and **b** test cases with parameter variations given in Table 1. The three cases presented correspond to $\varepsilon = 1 \times 10^{-2}$ and $K = 3$ (blue squares), $\varepsilon = 1 \times 10^{-3}$ and $K = 5$ (orange triangles) and $\varepsilon = 1 \times 10^{-4}$ and $K = 7$ (green points). Errors are computed as the maximum distance between two corresponding nodes and shown relative to the reference diameter, d_0
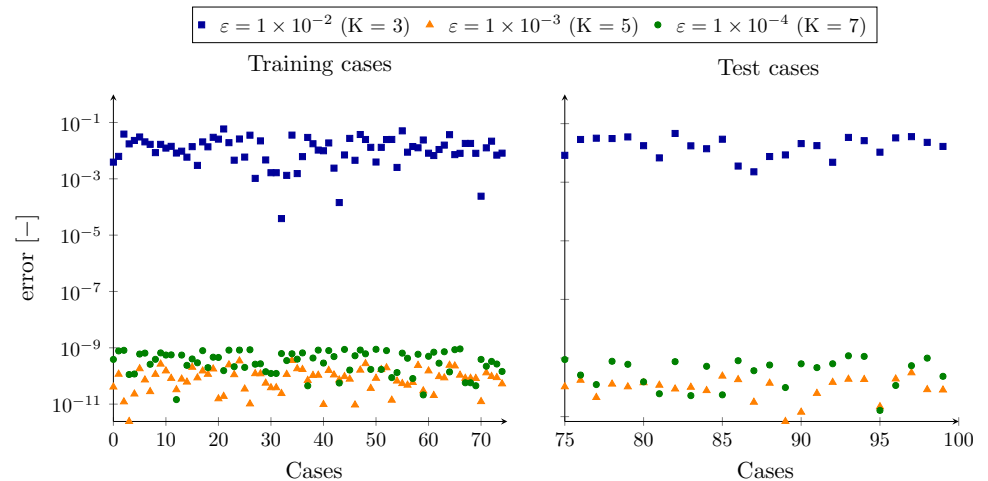
the ROM is set to 1.2 ms^{-1} to fully cover hyperaemic blood flow conditions, which need to be considered when assessing the pressure drop across the stenosis. Each FOM is initialized with a velocity field equal to the mean inlet velocity and calculations are conducted for $t \in [0, 0.1]$ s with a time step size of, $\Delta t = 5 \times 10^{-6}$ s and 2 correction iterations of the PISO algorithm. Snapshots of pressure and velocity are stored from $t = 0.08$ at intervals of $\Delta t = 5 \times 10^{-4}$ s to remove the influence of transients due to the initialization and focus on the steady-state/periodic solution which will then be used to compute the mean pressure drop. Since the FOM uses a splitting approach, we also stored the intermediate velocity and pressure from the inner iterations of the PISO algorithm to reduce the error propagation and enhance stability in the ROM. In total, 6150 snapshots for both velocity and pressure are used for the training of the ROM.

POD is then applied to the snapshots for the calculation of velocity and pressure subspaces as described in Sect. 2.5.1. The tolerance selected for both the velocity and pressure fields is 10^{-4} resulting in $N_v = 330$ and $N_p = 282$ modes, respectively. For the velocity we have considered an inner energy product by using the mass matrix of the mean geometry as the norm, i.e., $\mathbb{X}_h = M_h$, while for the pressure POD we have set $\mathbb{X}_h = I_h$ (Quarteroni et al. 2016).

Once the POD modes have been calculated, we solve the ROM Eq. (8) for each training case to approximate the hemodynamic solutions obtained by the FOM. Simulations are conducted for $t \in [0, 0.1]$ s with $\Delta t = 5 \times 10^{-6}$ s and 2 correction iterations of the PISO algorithm. The matrices of the algebraic description of the NS operators are saved for $t \in [0, 0.1]$ s at intervals of $\Delta t = 2.5 \times 10^{-3}$ s. In total, 3000 snapshots are stored for solution-dependent operators and 75 for the other NS operators. For the right-hand side of the pressure-Poisson equation, $\boldsymbol{\Delta}_h (\mathbf{u}_h)^{BC_N}$, we have also stored the intermediate operators obtained during the PISO inner iteration, leading to a total of 6000 snapshots available for the training.

DEIM is then applied to all operators and source terms obtained from the ROM with tolerances and total number of components reported in Table 2. The table also reports the number of bases and coefficients that allow reconstruction of the NS algebraic operators within the selected tolerance. Each coefficient can be calculated from a specific component of the algebraic matrix representation of the operator. By mapping the relative degrees of freedom to the corresponding mesh nodes, it is possible to identify the reduced mesh, which is a subset of the original domain required for the calculations of the DEIM coefficients. Therefore, since when using the HROM we are

Table 2 DEIM tolerances and number of bases retained for NS operators of Eq. (5)

Operator	ε	Bases
Momentum conservation equations		
$\mathbf{C}_N(\mathbf{u}_h)$	10^{-8}	1443
$\nu\mathbf{\Delta}_N$	10^{-8}	38
$\mathbf{\nabla}_N$	10^{-8}	33
$\mathbf{M}_N$	10^{-8}	16
$\mathbf{C}_N(\mathbf{u}_h)^{BC_D}$	10^{-8}	2
$\nu\mathbf{\Delta}_N^{BC_D}$	10^{-8}	2
$\nu\mathbf{\Delta}_N^{BC_N}$	10^{-8}	1519
Pressure-Poisson equations		
$\mathbf{\Delta}_N(\mathbf{u}_h)$	10^{-8}	697
$\mathbf{\Delta}_N(\mathbf{u}_h)^{BC_N}$	10^{-8}	2001

For the momentum conservation equations, $\mathbf{C}_N(\mathbf{u}_h)$ and $\mathbf{C}_N(\mathbf{u}_h)^{BC_D}$ are the algebraic form of the convective operator and the corresponding discretization of the Dirichlet boundary conditions, respectively; $\nu\mathbf{\Delta}_N$, $\nu\mathbf{\Delta}_N^{BC_D}$ and $\nu\mathbf{\Delta}_N^{BC_N}$, are the algebraic form of the Laplacian operator and the terms deriving from the discretization of the Dirichlet and Neumann boundary conditions, respectively; $\mathbf{\nabla}_N$ and $\mathbf{M}_N$ are the algebraic form of the pressure-gradient and inertia terms. In the pressure-Poisson equations, $\mathbf{\Delta}_N(\mathbf{u}_h)$ and $\mathbf{\Delta}_N(\mathbf{u}_h)^{BC_N}$ are the algebraic form of the Laplacian and the right-hand-side of the equations, respectively

only required to assemble the algebraic NS operators for the cells of the reduced mesh, during the *online* phase we use the geometric parametrization to reconstruct the cells of the reduced mesh, decreasing the computational cost compared to the reconstruction of the full mesh. In Fig. 5, panels a to d, we show the reduced mesh for the convective, Laplacian, pressure-gradient and pressure-Laplacian operators.

We chose to perform DEIM on the ROM operators rather than on those of the FOM to reduce the propagation of numerical errors when reconstructing the source terms of the momentum and pressure equations. In fact, the approximation of the solution with the RB of velocity and pressure introduces an error that is amplified by the derivatives in the formulation of the algebraic terms. By using the algebraic operators obtained from the ROM, this error is eliminated since the terms are built only from the bases used for the projection.

Finally, we used HROM to predict the pressure drop in the geometries of the database for selected inlet flow velocities. The time integration method is the same as for the ROM. Figure 6a shows the time-averaged pressure drop, ΔP, in mmHg predicted by the FOM (horizontal axis) and ROM and HDEIM (vertical axis; circles and triangles, respectively). This is to compare the approximation errors introduced by the Galerkin projection and by the reconstruction

(a)   **(b)**   **(c)**   **(d)**

Fig. 5 Reduced mesh for NS operators of the HROM. The panels represent the cells of the full mesh (red elements) required for the calculation of the coefficients used to impose the interpolation constraints on DEIM. The background geometry refers to the mean geometry obtained from the geometric parametrization (transparent background). The cases refer to the algebraic description of the **a** convective operator, $\mathbf{C}_N(\mathbf{u}_h)$, **b** Laplacian operator, $\nu\mathbf{\Delta}_N$, **c** pressure-gradient operator, $\mathbf{\nabla}_N$ and **d** pressure-Laplacian operator $\mathbf{\Delta}_N(\mathbf{u}_h)$

Fig. 6 Comparison of time-averaged pressure drop predictions from FOM, ROM and HROM for the training set. **a** compares the pressure drop determined with the FOM (horizontal axis) to that calculated with the ROM (green dots) and the HROM (orange triangles), while **b** shows the relative errors in the pressure predictions: e_R (green dots) refers to the relative error between FOM and ROM, while e_H (orange triangles) to the relative error between FOM and HROM
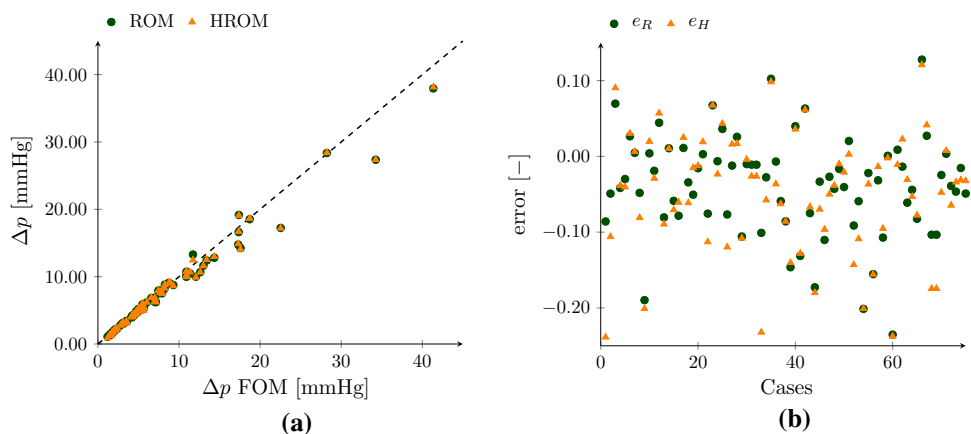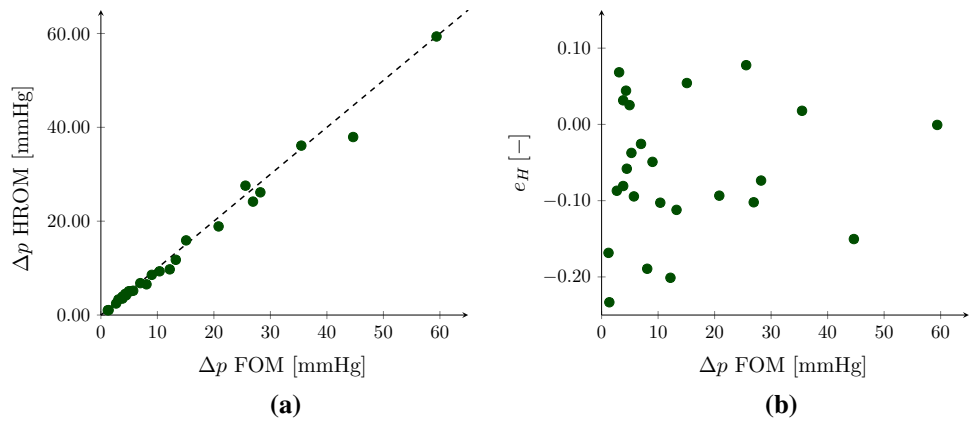
(a)

(b)

(a)

(b)

of the operators using DEIM. Figure 6b shows the error for each case relative to the corresponding full-order model. Specifically, e_R refers to the relative error in the pressure drop prediction between FOM and ROM, while e_H refers to the relative error in the pressure drop prediction between FOM and HROM. Time-averaged quantities are calculated in the time range [0.04–0.10] s.

Figure 7 compares the pressure predictions between ROM and HROM for the 25 test cases not included in the training (Fig. 7a) and the relative error, e_H, is plotted as function of the FOM pressure drop in Fig. 7b.

In Figs. 8 and 9 we show the time-averaged velocity magnitude and pressure fields for two cases with similar geometry but different physical parameters, yielding section-averaged inlet velocities of $0.34\,\mathrm{ms}^{-1}$ and $1.05\,\mathrm{ms}^{-1}$, respectively . We refer to the reconstructed velocity and pressure fields from HROM as $\mathbf{u}_r$ and p_r, respectively, which, in the spirit of the POD decomposition, are obtained as $\mathbf{u}_r = \mathbf{\Phi}_u \mathbf{u}_{N_v}$ and $p_r = \mathbf{\Phi}_p p_{N_p}$. In the plots, we compare the predictions of FOM and HROM and show the relative errors. Pressure fields shown in the respective right panels of Figs. 8 and 9 contain a region downstream of the stenosis with negative
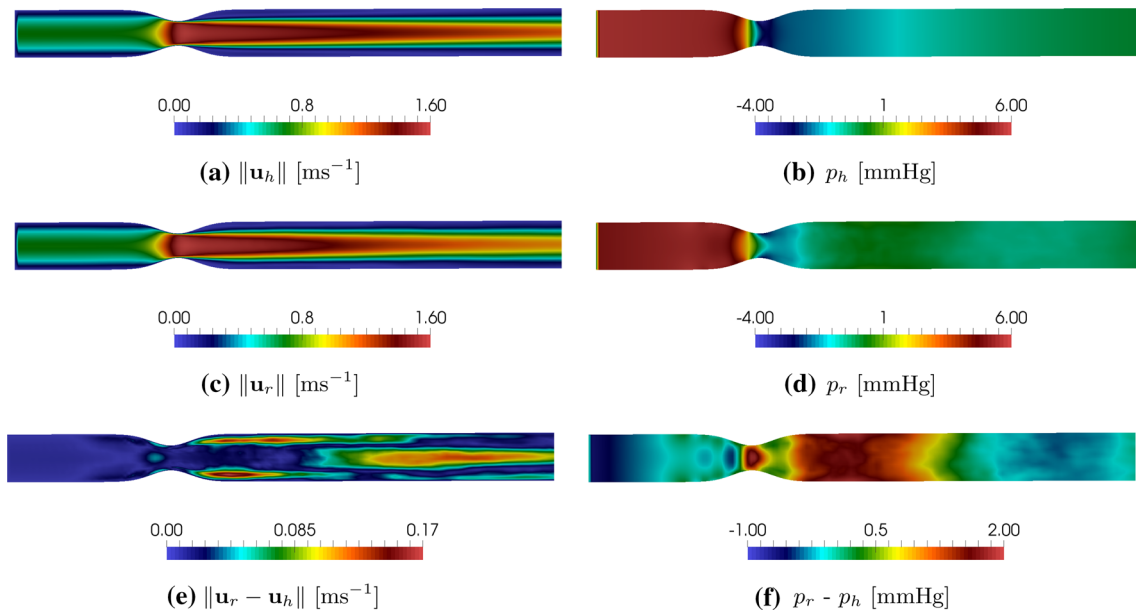

(a) $\|\mathbf{u}_h\|$ $[\mathrm{ms}^{-1}]$

(b) p_h $[\mathrm{mmHg}]$

(c) $\|\mathbf{u}_r\|$ $[\mathrm{ms}^{-1}]$

(d) p_r $[\mathrm{mmHg}]$

(e) $\|\mathbf{u}_r - \mathbf{u}_h\|$ $[\mathrm{ms}^{-1}]$

(f) $p_r - p_h$ $[\mathrm{mmHg}]$

Fig. 8 Contours of time-averaged velocity (left panels) and pressure fields (right panels) for the case described by the parameters $\mu_{g,0} = 3.4\,\mathrm{mm}$, $\mu_{g,1} = 0.48$, $\mu_{g,2} = 1.1 \times 10^{-4}$, $\mu_{g,3} = 0.47$, $\mu_{g,4} = 0.97$ and $\mu_p = 0.34\,\mathrm{ms}^{-1}$. The time-averaged pressure drops predicted by FOM and HROM are 5.69 mmHg and 5.59 mmHg, respectively. The prediction error e_H is thus 0.1 mmHg (2%). The maximum error in the pressure field reconstruction is approximately 30%. **a** and **b** Respectively, velocity magnitude, $\|\mathbf{u}_h\|$, and pressure, p_h, predictions from the FOM. **c** and **d** Respectively, the reconstructed velocity magnitude field, $\|\mathbf{u}_r\|$, and pressure, p_r, predictions from the HROM. Finally, panels **e** and **f** illustrate the absolute error in velocity and pressure of the HROM with respect to the FOM, showing the magnitude of velocity difference, $\|\mathbf{u}_r - \mathbf{u}_h\|$, and difference in pressure, $p_r - p_h$, respectively
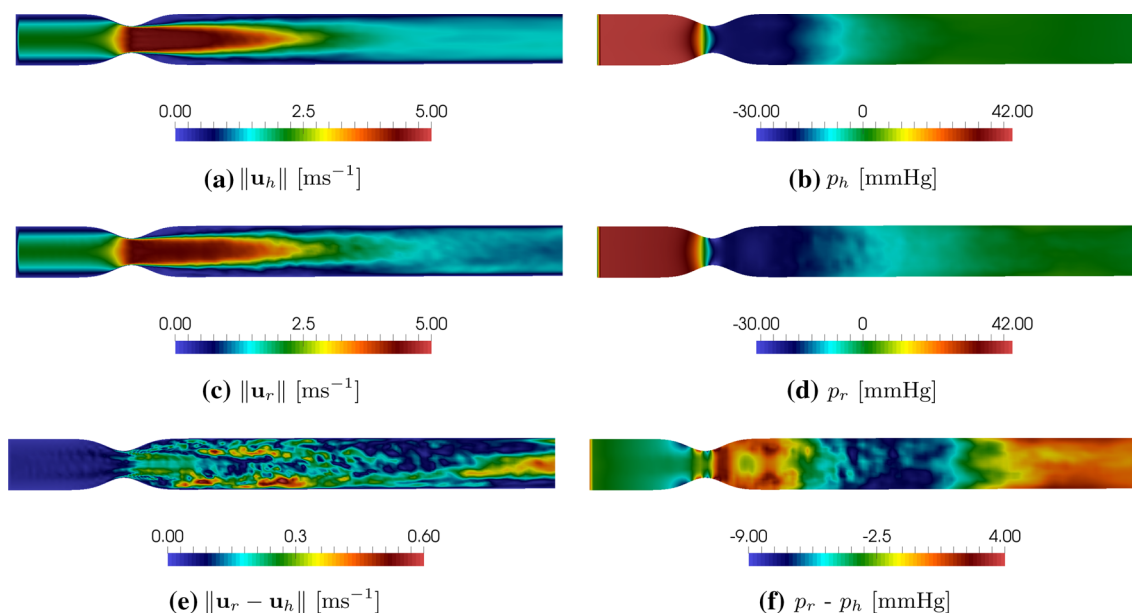
(a) $\|\mathbf{u}_h\|$ [ms^{-1}]

(b) p_h [mmHg]

(c) $\|\mathbf{u}_r\|$ [ms^{-1}]

(d) p_r [mmHg]

(e) $\|\mathbf{u}_r - \mathbf{u}_h\|$ [ms^{-1}]

(f) p_r - p_h [mmHg]

Fig. 9 Contours of time-averaged velocity (left panels) and pressure fields (right panels) for the case described by the parameters $\mu_{g,0} = 3.0$ mm, $\mu_{g,1} = 0.48$, $\mu_{g,2} = 0.75 \times 10^{-4}$, $\mu_{g,3} = 0.04$, $\mu_{g,4} = 0.93$ and $\mu_p = 1.05$ ms^{-1}. The time-averaged pressure drops predicted by FOM and HROM are 41.74 mmHg and 38.68 mmHg, respectively. The prediction error e_H is thus 3 mmHg (8%). The maximum error in the pressure field reconstruction is approximately 10%.

a and **b** Respectively, velocity magnitude, $\|\mathbf{u}_h\|$, and pressure, p_h, predictions from the FOM. **c** and **d** Respectively, the reconstructed velocity magnitude field, $\|\mathbf{u}_r\|$, and pressure, p_r, predictions from the HROM. Finally, **e** and **f** illustrate the absolute error in velocity and pressure of the HROM with respect to the FOM, showing the magnitude of velocity difference, $\|\mathbf{u}_r - \mathbf{u}_h\|$, and difference in pressure, $p_r - p_h$, respectively

pressure values. We emphasize that these values are relative to zero pressure prescribed at the outlet and indicate that after the stenosis there is a partial recovery of the pressure which then approaches zero.

4 Discussion

In this study, we have presented a novel computational approach based on a parametrized reduced-order model of the Navier–Stokes and continuity equations for accelerating the calculation of pressure drop along blood vessels. This is relevant for the functional assessment of coronary artery stenoses. The computational speedup is obtained by addressing two aspects of the standard pipeline for flow modeling: mesh generation and hemodynamic calculations (steps C and D of Fig. 1, respectively).

We have considered a realistic set of synthetic geometries with a geometric variability representing the most important anatomic features of coronary arteries such as diameter scaling and tapering, stenosis severity, position and length. Itu et al. (2016) have prescribed a similar set of geometric parameters to generate synthetic domains to compute the hemodynamic solution in a simplified one-dimensional Navier–Stokes model. Our approach is different in that we utilize the parametrization simply as a means to control the

variability of the shapes used to populate the database. Our geometric parametrization method can, in fact, identify automatically a suitable mapping based on the geometries provided in the database, allowing for an easy adaptation to general shapes for which a parametric description is not known or imposed a priori. This approach allows to reconstruct the computational domain in vessels of interest from control points on its surface. To our knowledge, this is the first example of geometric parametrization and reconstruction able to handle arbitrary shape variations based on DEIM. Furthermore, in contrast to approaches based on neural networks, our method derives directly from the fundamental laws of fluid mechanics as described by the incompressible Navier–Stokes equations rather than emulating certain characteristics of fluid dynamics.

We have shown that DEIM recovers the correct mapping using a number of modes comparable with the number of parameters selected for the generation of the anatomic variability. The magic points are clustered where the largest deformations are found. Mesh generation with the conventional pipeline takes on average 2 s on a single CPU core (Intel(R) Xeon(R) E5-1630), but only 0.19 s with the proposed geometric reconstruction, providing a speedup of about a factor of 10. Of the 0.19 s, 1.5 ms are required, on average, for computing the mesh coordinates, while the remaining time is overhead for writing the point coordinates

to file. This time intensive data exchange via files could be circumvented by coupling our code to OpenFOAM.

The parametric HROM is derived from a FOM that uses the OpenFOAM FV framework to discretize and solve the NS equations. Two aspects contribute to the novelty of the method: the fully nonlinear parametrization in a FV approximation and the pressure–velocity coupling of the NS equations using the PISO splitting scheme. Another standout feature of our approach compared to the state-of-the-art is automatically satisfied ROM stability. Earlier models based on FEM required extra terms to both recover the pressure–velocity coupling and ensure system stability. Our methodology reduces computational cost substantially: each time step requires the solution of a system of 330 and 282 equations for velocity and pressure field, respectively, compared to 225,000 and 75,000, respectively, for the FOM. Considering the computational time for the generation of the time-dependent operators and the solution of the system of equations with PISO, the FOM requires, on average, 2.5 s on a single core for each time step, while the ROM requires 0.11 s on average. This corresponds to a speedup of a factor of about 25. The times reported here include overhead for data exchange, which, as mentioned before, could be circumvented by coupling our code to OpenFOAM.

The errors observed between FOM and ROMs are linked to the complexity of the flow structures, as illustrated in Figs. 8 and 9. These two cases also show that the prediction errors in the pressure drop, e_H, is lower than the local maximum errors in the pressure fields, indicating that the output variable of interest is better approximated by the HROM than the full field. In Fig. 6 we have shown that the maximum prediction errors e_R and e_H are bounded by 25%, with a maximum absolute value below 8 mmHg. We observe that these two errors are very close, indicating negligible additional errors in the HROM introduced by the approximation of the NS operators with DEIM. We also find that the ROM and HROM mostly underestimate the pressure drop. A possible reason for this may be the fact that we neglect the smallest flow structures with our POD basis, and therewith also some of the inherent viscous dissipation. The errors e_H for training and test sets are very similar (Figs. 6a, 7a), indicating that the model could be used for making predictions for new cases not included in the training set. In addition, the magnitude of these errors is within the range of those introduced by the catheter wire during invasive FFR measurements (Ashtekar et al. 2007; de Vecchi et al. 2014), showing the potential of the method for the generation of reduced-order models fast enough for pressure drop predictions for clinical applications. The approximation error e_H, which is the combined result of approximations of geometry, Navier–Stokes operators and POD modes, could be reduced by increasing the number of POD bases considered. However, this can quickly escalate the computational cost. In fact, we need to point out that while in the FOM a sparse system of equations needs to be solved, the system matrices in the ROM are dense. The optimization of accuracy and speedup will be fundamental in future clinical applications. Indeed, we have shown that errors are not uniformly distributed in the full range of pressure drop predictions. High prediction accuracy must be achieved in cases where the FFR value is in the range of 0.75–0.85, since 0.8. is commonly used as the threshold below which a stenosis is considered functionally significant. We have given the same weight to all solution snapshots used in the POD and DEIM calculations. For clinical applications, the cases close to the FFR threshold value of 0.8 should receive a higher weight, and would thus increase the accuracy of the predictions in that FFR region.

5 Conclusion

We have presented a methodology for generating a parametrized reduced-order model (HROM) to predict the pressure drop along a blood vessel. We have used an *offline—online* splitting of the solution process combined with proper orthogonal decomposition (POD) and the discrete empirical interpolation method (DEIM) to reduce the computational cost associated with mesh generation and the solution of the hemodynamic equations. The computational domain inside the lumen is defined from the assembly of precomputed geometric bases weighted by corresponding coefficients computed from the coordinates of a few sampling points on the vessel wall. In our example case, this has led to a speedup of about a factor of 10 compared to the conventional mesh generation process. This parametrization works with arbitrary shape variations, and without the need for defining ad hoc functions or mappings. To our knowledge, this is the first application of such a flexible approach for the generation of a parametric computational domain that does not require an analytical description of the anatomical variations and that can be directly coupled with the reconstruction method of the NS algebraic operators. A more substantial reduction in computational cost is achieved with the HROM for the incompressible Navier–Stokes equations, which can, principally, be adapted to any vessel of the body. To our knowledge, it is the first application of a complete parametric nonlinear reduction approach to the finite volume description of the Navier–Stokes equations using open-source libraries. Numerical results show good overall accuracy of the HROM in predicting hemodynamic indices, and a speedup of about a factor of 25 compared to the full-order model. Errors are lower than those incurred during invasive acquisitions of pressure drop across stenoses in coronary arteries. Altogether, the methodology described

herein may allow for the development of a clinically viable framework for the noninvasive assessment of the functional severity of arterial stenoses.

Compliance with ethical standards

Conflict of interest A patent application covering parts of the technology described in the manuscript has been filed.

References

Amsallem D, Zahr M, Farhat C (2012) Nonlinear model order reduction based on local reduced-order bases. Int J Numer Methods Eng 92(10):891–916

Ashtekar KD, Back LH, Khoury SF, Banerjee RK (2007) In vitro quantification of guidewire flow-obstruction effect in model coronary stenoses for interventional diagnostic procedure. ASME J Med Dev 1(3):185–196

Ballarin F, Manzoni A, Quarteroni A, Rozza G (2015) Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. Int J Numer Methods Eng 102(5):1136–1161

Ballarin F, Faggiano E, Ippolito S, Manzoni A, Quarteroni A, Rozza G, Scrofani R (2016) 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 315:609–628

Ballarin F, Faggiano E, Manzoni A, Quarteroni A, Rozza G, Ippolito S, Antona C, Scrofani R (2017) Numerical modeling of hemodynamics scenarios of patient-specific coronary artery bypass grafts. Biomech Model Mechanobiol 16(4):1373–1399

Bergmann M, Bruneau CH, Iollo A (2009) Enablers for robust POD models. J Computat Phys 228(2):516–538

Buljak V (2011) Inverse analyses with model reduction. Computational fluid and solid mechanics. Springer, Berlin

Buoso S, Palacios R (2017) On-demand aerodynamics of integrally actuated membranes with feedback control. AIAA J 55(2):377–388

Coenen A, Lubbers MM, Kurata A, Kono A, Dedic A, Chelu RG, Dijkshoorn ML, Gijsen FJ, Ouhlous M, van Geuns RM, Nieman K (2015) Fractional flow reserve computed from noninvasive CT angiography data: diagnostic performance of an on-site clinician-operated computational fluid dynamics algorithm. Radiology 274(3):674–683

Colciago C, Deparis S, Quarteroni A (2014) Comparisons between reduced order models and full 3D models for fluid-structure interaction problems in haemodynamics. J Comput Appl Math 265:20–138

de Vecchi A, Clough RE, Gaddum NR, Rutten MCM, Lamata P, Schaeffter T, Nordsletten DA, Smith NP (2014) Catheter-induced errors in pressure measurements in vessels: an in-vitro and numerical study. IEEE Trans Biomed Eng 61(6):1844–1850

de Zélicourt DA, Kurtcuoglu V (2016) Patient-specific surgical planning, where do we stand? The example of the Fontan procedure. Ann Biomed Eng 44(1):174–186

Douglas PS, De Bruyne B, Pontone G, Patel MR, Norgaard BL, Byrne RA, Curzen N, Purcell I, Gutberlet M, Rioufol G, Hink U, Schuchlenz HW, Feuchtner G, Gilard M, Andreini D, Jensen JM, Hadamitzky M, Chiswell K, Cyr D, Wilk A, Wang F, Rogers C, Hlatky MA (2016) 1-Year outcomes of FFRCT-guided care in patients with suspected coronary disease. J Am Coll Cardiol 68(5):435–445

Gijsen FJ, Schuurbiers JC, van de Giessen AG, Schaap M, van der Steen AF, Wentzel JJ (2014) 3D reconstruction techniques of human coronary bifurcations for shear stress computations. J Biomech 47(1):39–43

Gould KL, Lipscomb K, Hamilton WG (1974) Physiologic basis for assessing critical coronary stenosis. Am J Cardiol 33(1):87–94

Heidenreich PA, Trogdon JG, Khavjou OA, Butler J, Dracup K, Ezekowitz MD, Finkelstein EA, Hong Y, Johnston SC, Khera A, Lloyd-Jones DM, Nelson SA, Nichol G, Orenstein D, Wilson PW, Woo YJ (2011) Forecasting the future of cardiovascular disease in the United States. Circulation 123(8):933–944

Hlatky MA, De Bruyne B, Pontone G, Patel MR, Norgaard BL, Byrne RA, Curzen N, Purcell I, Gutberlet M, Rioufol G, Hink U, Schuchlenz HW, Feuchtner G, Gilard M, Andreini D, Jensen JM, Hadamitzky M, Wilk A, Wang F, Rogers C, Douglas PS (2015) Quality-of-life and economic outcomes of assessing fractional flow reserve with computed tomography angiography. J Am Coll Cardiol 66(21):2315–2323

Itu L, Rapaka S, Passerini T, Georgescu B, Schwemmer C, Schoebinger M, Flohr T, Sharma P, Comaniciu D (2016) A machine-learning approach for computation of fractional flow reserve from coronary computed tomography. J Appl Physiol 121:42–52

Jasak H (1996) Error analysis and estimation for the finite volume method with applications to fluid flows. Ph.D. thesis, Imperial College London

Keegan J, Gatehouse PD, Yang G-Z, Firmin DN (2004) Spiral phase velocity mapping of left and right coronary artery blood flow: correction for through-plane motion using selective fat-only excitation. J Magn Reson Imaging 20(6):953–960

Knight J, Olgac U, Saur SC, Poulikakos D, Marshall W, Cattin PC, Alkadhi H, Kurtcuoglu V (2010) Choosing the optimal wall shear parameter for the prediction of plaque location: a patient-specific computational study in human right coronary arteries. Atherosclerosis 211(2):445–450

Lassila T, Manzoni A, Quarteroni A, Rozza G (2013) A reduced computational and geometrical framework for inverse problems in haemodynamics. Int J Numer Methods Biomed Eng 29(7):741–776

Maday Y, Nguyen NC, Patera TA, Pau SH (2009) A general multipurpose interpolation procedure: the magic points. Commun Pure Appl Anal 8:383

Mancini GBJ, Ryomoto A, Kamimura C, Yeoh E, Ramanathan K, Schulzer M, Hamburger J, Ricci D (2007) Redefining the normal angiogram using population-derived ranges for coronary size and shape: validation using intravascular ultrasound and applications in diverse patient cohorts. Int J Cardiovasc Imaging 23(4):441–453

Manzoni A (2014) An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier–Stokes flows. ESAIM Math Model Numer Anal 48:1199–1226

Manzoni A, Quarteroni A, Rozza G (2012a) Model reduction techniques for fast blood flow simulation in parametrized geometries. Int J Numer Methods Biomed Eng 28(6–7):604–625

Manzoni A, Quarteroni A, Rozza G (2012b) Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques. Int J Numer Methods Fluids 70(5):646–670

Moukalled F, Mangani L, Darwish M (2015) The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM and Matlab, 1st edn. Springer, Berlin

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

Olgac U, Poulikakos D, Saur SC, Alkadhi H, Kurtcuoglu V (2009) Patient-specific three-dimensional simulation of LDL accumulation in a human left coronary artery in its healthy and atherosclerotic states. Am J Physiol Heart Circ Physiol 296(6):H1969–H1982

Pagani S, Manzoni A, Quarteroni A (2018) Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. Comput Methods Appl Mech Eng 340:530–558

Pijls NH, van Schaardenburgh P, Manoharan G, Boersma E, Jan-Willem B, vant Veer M, Bär F, Hoorntje J, Koolen J, Wijns W, de Bruyne B (2007) Percutaneous coronary intervention of functionally nonsignificant stenosis. J Am Coll Cardiol 49(21):2105–2111

Quarteroni A, Manzoni A, Negri F (2016) Reduced basis methods for partial differential equations: an introduction. Vol. 92 of UNI-TEXT - La Matematica per il 3+2. Springer, Berlin

Quarteroni A, Rozza G (2007) Numerical solution of parametrized Navier–Stokes equations by reduced basis methods. Numer Methods Partial Differ Equ 23(4):923–948

Rikhtegar F, Knight JA, Olgac U, Saur SC, Poulikakos D, Marshall W, Cattin PC, Alkadhi H, Kurtcuoglu V (2012) Choosing the optimal wall shear parameter for the prediction of plaque locationa patient-specific computational study in human left coronary arteries. Atherosclerosis 221(2):432–437

Rowley CW (2011) Model reduction for fluids, using balanced proper orthogonal decomposition. Int J Bifurc Chaos 15(03):997–1013

Sankaran S, Esmaily Moghadam M, Kahn AM, Tseng EE, Guccione JM, Marsden AL (2012) Patient-specific multiscale modeling of blood flow for coronary artery bypass graft surgery. Ann Biomed Eng 40(10):2228–2242

Stabile G, Hijazi S, Mola A, Lorenzi S, Rozza G (2015) Advances in reduced order modelling for CFD: vortex shedding around a circular cylinder using a POD-Galerkin method. Commun Appl Ind Math 9(1):1–s21

Stabile G, Rozza G (2018) Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier Stokes equations. Comput Fluids 173:273–284

Stergiopulos N, Meister JJ, Westerhof N (1995) Evaluation of methods for estimation of total arterial compliance. Am J Physiol Heart Circ Physiol 268(4):H1540–H1548

Taylor CA, Fonte TA, Min JK (2013) Computational fluid dynamics applied to cardiac computed tomography for noninvasive quantification of fractional flow reserve. J Am Coll Cardiol 61(22):2233–2241

Tonino PA, Fearon WF, Bruyne BD, Oldroyd KG, Leesar MA, Lee PNV, MacCarthy PA, van't Veer M, Pijls NH (2010) Angiographic versus functional severity of coronary artery stenoses in the FAME study. J Am Coll Cardiol 55(25):2816–2821

Vergallo R, Papafaklis MI, Yonetsu T, Bourantas CV, Andreou I, Wang Z, Fujimoto JG, McNulty I, Lee H, Biasucci LM, Crea F, Feldman CL, Michalis LK, Stone PH, Jang I-K (2014) Endothelial shear stress and coronary plaque characteristics in humans. Circ Cardiovasc Imaging 7(6):905–911

Weller HG, Tabor G, Jasak H, Fureby C (1998) A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 12(6):620–631

Zafar H, Sharif F, Leahy MJ (2014) Measurement of the blood flow rate and velocity in coronary artery stenosis using intracoronary frequency domain optical coherence tomography: validation against fractional flow reserve. Int J Cardiol Heart Vasc Supplement C(5):68–71

Zhang JM, Zhong L, Luo T, Lomarda AM, Huo Y, Yap J, Lim ST, Tan RS, Wong ASL, Tan JWC, Yeo KK, Fam JM, Keng FYJ, Wan M, Su B, Zhao X, Allen JC, Kassab GS, Chua TSJ, Tan SY (2016) Simplified models of non-invasive fractional flow reserve based on CT images. PLoS ONE 11(5):1–20

Zimmermann FM, Ferrara A, Johnson NP, van Nunen LX, Escaned J, Albertsson P, Erbel R, Legrand V, Gwon H-C, Remkes WS, Stella PR, van Schaardenburgh P, Jan Willem G, De Bruyne B, Pijls NH (2015) Deferral vs. performance of percutaneous coronary intervention of functionally non-significant coronary stenosis: 15-year follow-up of the DEFER trial. Eur Heart J 36(45):3182–3188