Check for updates

Received: 6 April 2023 | Revised: 2 August 2023 | Accepted: 3 August 2023

DOI: 10.1002/nme.7342

RESEARCH ARTICLE

WILEY

Model order reduction for deforming domain problems in a time-continuous space-time setting

Fabian Key[1] | **Max von Danwitz**[2] | **Francesco Ballarin**[3] | **Gianluigi Rozza**[4]

[1]Institute of Lightweight Design and Structural Biomechanics (ILSB), TU Wien, Vienna, Austria

[2]Institute for Mathematics and Computer-Based Simulation (IMCS), Universität der Bundeswehr München, Neubiberg, Germany

[3]Department of Mathematics and Physics, Università Cattolica del Sacro Cuore, Brescia, Italy

[4]Mathematics Area, mathLab, Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy

Correspondence
Fabian Key, Institute of Lightweight Design and Structural Biomechanics (ILSB), TU Wien, Karlsplatz 13, 1040 Vienna, Austria.
Email: fabian.key@tuwien.ac.at

Funding information
European Union–NextGenerationEU; Gruppo Nazionale per il Calcolo Scientifico; Marie Sklodowska-Curie Actions, Grant/Award Number: 872442; Technische Universität Wien Bibliothek; Università Cattolica del Sacro Cuore, Grant/Award Number: CUP_E53C22001930001

Abstract

In the context of simulation-based methods, multiple challenges arise, two of which are considered in this work. As a first challenge, problems including time-dependent phenomena with complex domain deformations, potentially even with changes in the domain topology, need to be tackled appropriately. The second challenge arises when computational resources and the time for evaluating the model become critical in so-called many query scenarios for parametric problems. For example, these problems occur in optimization, uncertainty quantification (UQ), or automatic control, and using highly resolved full-order models (FOMs) may become impractical. To address both types of complexity, we present a novel projection-based model order reduction (MOR) approach for deforming domain problems that takes advantage of the time-continuous space-time formulation. We apply it to two examples that are relevant to engineering or biomedical applications and conduct an error and performance analysis. In both cases, we are able to drastically reduce the computational expense for a model evaluation and, at the same time, maintain an adequate accuracy level, each compared to the original time-continuous space-time full-order model (FOM). All in all, this work indicates the effectiveness of the presented MOR approach for deforming domain problems by taking advantage of a time-continuous space-time setting.

KEYWORDS

deforming domain problems, finite element method, model order reduction, time-continuous space-time approach

1 | INTRODUCTION

The use of simulation-based methods is nowadays widespread in scientific and engineering applications. Commonly, these methods are intended to serve purposes like enhanced insight and prediction capabilities with respect to the processes under investigation. In this way, they can be useful tools in the context of product design or optimization procedures. Furthermore, they may also help to quantify or enhance the reliability of a given process by techniques such as uncertainty quantification (UQ). Lastly, they further allow to be integrated into ongoing operations, for example, as digital twins or to set up optimal control.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2023 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

Arising in the context of simulation-based methods, we will focus on two specific types of complexity in this work. First, we consider applications that show a *transient* behavior involving a *deforming domain* and, potentially, even *topology changes*. These aspects need to be treated appropriately by the computational approach to cover all the relevant effects. In the following, we restrict ourselves to methods that make use of a computational mesh although there exists a variety of further simulation methods. In the case discussed here, the deformation of the domain requires methods that can appropriately handle both the computational mesh and the unsteady solution field. Existing methods will be discussed in more detail in Section 3.1. The second kind of complexity is related to the *expense*—in terms of computational resources and time—needed for an evaluation of the computational model. Especially the design and development phase of a process or a product may entail the assessment of various operating points or configurations, the optimization of process settings or component features, as well as uncertainty regarding involved quantities. In all cases, one ends up with a so-called many query scenario in which a great number of model evaluations needs to be performed. Furthermore, the integration into an automatic control environment may demand fast feedback from the simulation model. Typically, one can formulate the involved problems in a parametric manner where each of the model evaluations is characterized by a certain sample of parameter values. In these situations, employing the original model, which is often referred to as *full-order model (FOM)*, for each sample may easily exceed available resources or required feedback times. Here, the application of model order reduction (MOR) can provide a remedy. Based on the original model, a *reduced-order model (ROM)* with decreased computational complexity is constructed, while keeping its accuracy in the desired range. Common MOR techniques will be presented later in Section 4.1.

The work presented here addresses both types of complexity described above and entails a MOR approach for problems that are characterized by a transient solution field in a deforming domain with possibly changing topology. In the following, we restrict ourselves to scenarios in which the deformation is known a-priori. Nevertheless, there exist a variety of relevant examples with prescribed motion in engineering, for example, the flow in an internal combustion engine or around a ship's propeller.

Regarding the question of deforming domain problems, we rely on the *time-continuous space-time* setting. To significantly decrease the computational demands and/or response times of the underlying model in a next step, we apply a *projection-based* MOR technique for which we make use of proper orthogonal decomposition (POD). The key of the proposed approach is the particular combination of this MOR technique with the choice of a time-continuous space-time formulation. This connection allows us to construct a corresponding ROM for the aforementioned class of problems in a straightforward manner. In this way, the benefits of MOR are made easily accessible even when dealing with deforming domain problems that involve an unsteady solution and, if necessary, changes in the spatial topology. Alternative MOR approaches that exist and are applicable to transient or deformation-driven scenarios will also be discussed later, that is, in Section 4.1.

As examples of the type of problems in focus, we consider the simulation of two specific transient fluid flow scenarios like they may appear in engineering or biomedical applications. The domain deformation in these examples results from a moving valve plug or the narrowing of flexible artery walls. Furthermore, parameterization is induced through a variation of the material properties of the fluid or of the boundary conditions.

The remainder of the article is structured as follows: in Section 2, we derive the parametric formulation for fluid flow problems in deforming domains using the space-time perspective. Next, Section 3 contains the description of the respective FOM, which will be based on the finite element method (FEM). Subsequently, the construction of the corresponding ROM using POD and projection is outlined in Section 4. Results for the two fluid flow test cases covering three- and four-dimensional space-time domains are presented in Section 5 to illustrate the aptitude of the approach introduced in this work. Finally, our findings are summarized and discussed in Section 6.

2 | PARAMETRIC PROBLEM FORMULATION FOR FLUID FLOW IN DEFORMING DOMAINS

To lay the foundation for the subsequent MOR approach, we begin by deriving the *parametric formulation* of transient flow problems defined in deforming domains. As mentioned before, parametric problems, for example, may occur in the context of UQ or automatic control. Specifically, we consider problems that can be parametric in the *material properties* involved or in their *boundary conditions*. The variations in the material illustrate potential uncertainties in the process under investigation, which may be analyzed using UQ. Apart from that, adjustable boundary conditions

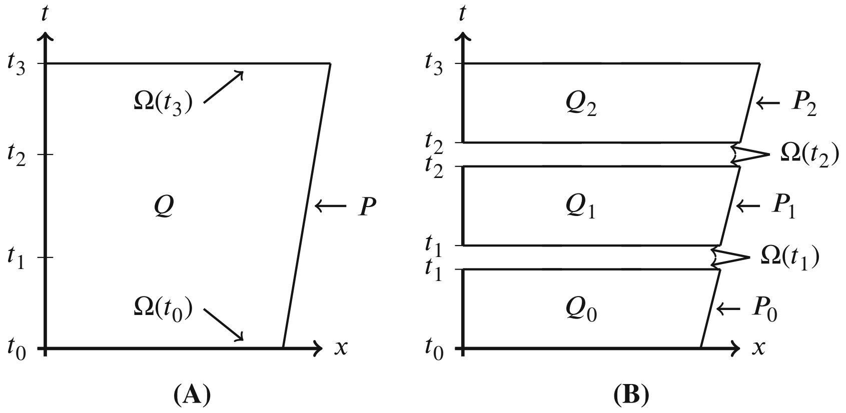


FIGURE 1 Definition of the computational domain(s) in the space-time approach for deforming domain problems. (A) The time-continuous approach. (B) The time-discontinuous approach.

prescribing, for example, the inflow velocity, showcase the type of use in a context such as automatic control. As mentioned earlier, we focus on unsteady processes that take place in time-varying domains, which potentially undergo topology changes, too.

For the description of the time-dependent solution field in a deforming domain, it is convenient to treat space and time coordinates, which are denoted by $\boldsymbol{x}$ and t, in the same way. To that end, the *time-continuous space-time domain* Q is introduced, which results from the time-dependent spatial domain $\Omega(t)$ and the time interval $[0, T]$, where n_{sd} describes the number of spatial dimensions:

$$Q = \left\{(\boldsymbol{x}, t) \in \mathbb{R}^{n_{\mathrm{sd}}} \times [0, T] \mid \boldsymbol{x} \in \Omega(t)\right\}, \tag{1}$$

such that $(\boldsymbol{x}, t) \in Q$. The respective space-time boundary is denoted as P where portions on which conditions of Dirichlet or Neumann type are prescribed are referred to as P_{D} and P_{N}, respectively. Furthermore, initial conditions can be imposed over $\Omega(t = 0) = \Omega(t_0)$. A sketch can be found in Figure 1A. In contrast, in a *time-discontinuous space-time* approach, one considers so-called time slabs Q_m with boundaries P_m (see Figure 1B). This results in a kind of time-stepping scheme with a modified computational domain in each time step. In that sense, the time-discontinuous approach is comparable to a semi-discrete description. A systematic comparison of time-continuous and time-discontinuous space-time formulations has been performed for advection-diffusion problems.[1] However, as we would like to stress again, the time-continuous description is essential to directly apply established MOR techniques to deforming domain problems later.

To be able to eventually demonstrate the presented approach, we consider a specific example of fluid flow equations here. In particular, we make use of the *Stokes equations*, which can be used to model a variety of fluid flows of low Reynolds numbers, for example. Based on the conservation of mass and momentum, the Stokes equations provide statements for the, in this case parametric, *velocity* and *pressure* fields of the fluid, which are referred to as $\boldsymbol{u}(\boldsymbol{x}, t; \boldsymbol{\mu})$ and $p(\boldsymbol{x}, t; \boldsymbol{\mu})$, respectively. Here, $\boldsymbol{\mu}$ denotes the *parameter vector* collecting parameter values to vary either material properties or boundary conditions. Note that we will drop the arguments of $\boldsymbol{u}$ and p for the sake of notation in the following. Nevertheless, dependencies on $\boldsymbol{\mu}$ will be stressed if necessary.

The resulting boundary value problem (BVP) in the space-time setting reads as follows:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}(\boldsymbol{\mu}) = 0 \qquad \text{in } Q, \tag{2}$$

$$\rho\left(\frac{\partial \boldsymbol{u}(\boldsymbol{\mu})}{\partial t} - \boldsymbol{f}\right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p; \boldsymbol{\mu}) = \boldsymbol{0} \qquad \text{in } Q, \tag{3}$$

$$\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{u}_{\mathrm{D}}(\boldsymbol{\mu}) \qquad \text{on } P_{\mathrm{D}}, \tag{4}$$

$$\boldsymbol{\sigma}(\boldsymbol{u}, p; \boldsymbol{\mu}) \cdot \boldsymbol{n} = \boldsymbol{h}(\boldsymbol{\mu}) \qquad \text{on } P_{\mathrm{N}}, \tag{5}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0; \boldsymbol{\mu}) = \boldsymbol{u}_0(\boldsymbol{\mu}) \qquad \text{in } \Omega(t_0). \tag{6}$$

Here, ρ is the fluid *density*, $\boldsymbol{f}$ is a *body force*, while $\boldsymbol{u}_{\mathrm{D}}(\boldsymbol{\mu})$, $\boldsymbol{u}_0(\boldsymbol{\mu})$, and $\boldsymbol{h}(\boldsymbol{\mu})$ are the parameter-dependent prescribed velocity values and a *traction vector*, respectively. Moreover, we consider parametric material properties through the fluid *viscosity* $\eta(\boldsymbol{\mu})$ that is included in the *Cauchy stress tensor* given by

$$\boldsymbol{\sigma}(\boldsymbol{u}, p; \boldsymbol{\mu}) = -p(\boldsymbol{\mu})\boldsymbol{I} + 2\eta(\boldsymbol{\mu})\boldsymbol{\epsilon}(\boldsymbol{u}(\boldsymbol{\mu})), \tag{7}$$

where $\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T)$ is the *rate-of-strain tensor*. In the following, we will consider fluid materials whose viscous properties cannot be appropriately described by a Newtonian model. Examples of such non-Newtonian fluids are plastics melt or blood. To account for shear-thinning effects of the fluid, we use the Carreau-Yasuda model[2-4] for the viscosity, which depends on the shear rate

$$\dot{\gamma}(\boldsymbol{u}; \boldsymbol{\mu}) = \sqrt{2\boldsymbol{\epsilon}(\boldsymbol{u}(\boldsymbol{\mu})) : \boldsymbol{\epsilon}(\boldsymbol{u}(\boldsymbol{\mu}))}. \tag{8}$$

To stress the dependency on the velocity field, we from now on write $\eta(\boldsymbol{u}; \boldsymbol{\mu})$ given by

$$\eta(\boldsymbol{u}; \boldsymbol{\mu}) = \eta_\infty + (\eta_0 - \eta_\infty)\left[1 + (\lambda\dot{\gamma})^a\right]^{\frac{n-1}{a}}, \tag{9}$$

with zero-shear-rate viscosity η_0, infinite-shear-rate viscosity η_∞, characteristic time λ, power-law index n, and a dimensionless parameter a describing the transition region between the zero-shear-rate viscosity and the power-law regions. In addition to the indirect parameter dependency of the viscosity through the parametric velocity field, we also take into account potential fluctuations in the model parameters. This is motivated by the fact that these model parameters can only be determined through a regression-based approach using measurement data associated with uncertainties.

Since the FEM will be used later to construct the FOM in Section 3, which in turn will serve as the basis for the ROM presented in Section 4, it is worthwhile to state the *weak formulation* of the problem here. We will assume that appropriate trial and weighting spaces for the velocity and the pressure field in the space-time domain Q are given. For the velocity, the trial and weighting spaces are denoted as $\boldsymbol{\mathcal{S}_u}(Q)$ and $\boldsymbol{\mathcal{V}_u}(Q)$, respectively. For the pressure, the trial space is also the weighting space and they are referred to as $\mathcal{S}_p(Q)$. The resulting weak formulation of the parametric problem defined in the space-time domain Q reads:

Find$(\boldsymbol{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in \boldsymbol{\mathcal{S}_u}(Q) \times \mathcal{S}_p(Q)$, *such that*$\forall(\boldsymbol{w}, q) \in \boldsymbol{\mathcal{V}_u}(Q) \times \mathcal{S}_p(Q)$:

$$\begin{aligned}&\int_Q \boldsymbol{w} \cdot \rho\left(\frac{\partial \boldsymbol{u}(\boldsymbol{\mu})}{\partial t} - \boldsymbol{f}\right) dQ - \int_Q p(\boldsymbol{\mu})(\boldsymbol{\nabla} \cdot \boldsymbol{w})\, dQ \\ &+ \int_Q \boldsymbol{\epsilon}(\boldsymbol{w}) : 2\eta(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{\epsilon}(\boldsymbol{u})\, dQ + \int_Q q(\boldsymbol{\nabla} \cdot \boldsymbol{u}(\boldsymbol{\mu}))\, dQ = \int_{P_{\mathrm{N}}} \boldsymbol{w} \cdot \boldsymbol{h}(\boldsymbol{\mu})\, dP.\end{aligned} \tag{10}$$

To simplify notation, the following short forms will be used in the remainder of this work:

$$\begin{aligned}\boldsymbol{u}_t(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{u}(\boldsymbol{\mu})}{\partial t}, & (\boldsymbol{w}, \boldsymbol{u}_t; \boldsymbol{\mu})_Q &= \int_Q \rho \boldsymbol{w} \cdot \boldsymbol{u}_t(\boldsymbol{\mu})\, dQ, \\ b(\boldsymbol{u}, q; \boldsymbol{\mu})_Q &= \int_Q q(\boldsymbol{\nabla} \cdot \boldsymbol{u}(\boldsymbol{\mu}))\, dQ, & a(\boldsymbol{w}, \boldsymbol{u}; \boldsymbol{v}, \boldsymbol{\mu})_Q &= \int_Q \boldsymbol{\epsilon}(\boldsymbol{w}) : 2\eta(\boldsymbol{v}, \boldsymbol{\mu})\boldsymbol{\epsilon}(\boldsymbol{u})\, dQ, \\ (\boldsymbol{w}, \boldsymbol{f})_Q &= \int_Q \rho \boldsymbol{w} \cdot \boldsymbol{f}\, dQ, & (\boldsymbol{w}, \boldsymbol{h}; \boldsymbol{\mu})_{P_{\mathrm{N}}} &= \int_{P_{\mathrm{N}}} \boldsymbol{w} \cdot \boldsymbol{h}(\boldsymbol{\mu})\, dP.\end{aligned}$$

In this notation, Equation (10) can be stated as

$$(\boldsymbol{w}, \boldsymbol{u}_t; \boldsymbol{\mu})_Q - b(\boldsymbol{w}, p; \boldsymbol{\mu})_Q + a(\boldsymbol{w}, \boldsymbol{u}; \boldsymbol{u}, \boldsymbol{\mu})_Q + b(\boldsymbol{u}, q; \boldsymbol{\mu})_Q = (\boldsymbol{w}, \boldsymbol{f})_Q + (\boldsymbol{w}, \boldsymbol{h}; \boldsymbol{\mu})_{P_{\mathrm{N}}}. \tag{11}$$

3 | FULL-ORDER MODEL FOR FLUID FLOW IN DEFORMING DOMAINS

In this section, we will derive the FOM which results from applying the FEM to the parametric problem introduced beforehand. In particular, this model will be capable of simulating transient problems in deforming domains. To that end, Section 3.1 will first contain a brief outline of existing techniques that address deforming domain problems in this context. Subsequently, we present the space-time FEM used in this work in Section 3.2.

3.1 | Methodological background for deforming domains in the full-order model

In the following, we will roughly outline existing approaches for deforming domain problems, that is, problems including moving boundaries or internal interfaces, without claiming completeness. First, one can make a division into interface capturing and interface tracking methods.[5] Well-known examples of interface capturing methods, which employ an implicit description of domain boundaries or interfaces on a background mesh, are the level-set method[6,7] or the volume-of-fluid method.[8] In contrast, interface tracking methods are based on boundary-conforming meshes. Thus, an update procedure that adapts the computational mesh according to the deforming domain is needed.

There exist a vast number of strategies, ranging from global remeshing to elaborate methods specifically designed for certain applications or types of deformation. As examples of connectivity-preserving methods that update nodal coordinates, one can mention partial differential equation (PDE)-based methods like the elastic mesh update method (EMUM),[9] spring-based methods,[10] or techniques using radial basis functions.[11] Further methods extend these strategies by a subsequent mesh optimization, for example, through edge swapping or vertex smoothing operations.[12,13] Moreover, specialized mesh update methods use algebraic operations to control the mesh evolution for a-priori known boundary deformations.[14,15] Furthermore, local remeshing strategies limit the cost of maintaining a high-quality boundary-conforming mesh under large deformations.[16-18] Broadening the scope of these methods, parts of the mesh can be activated and deactivated[19] and topology changes of the computational domain can be handled elegantly.[20]

As an alternative, one can also follow weak domain coupling strategies to account for the moving boundary or interface by using composite grids and introducing additional conditions on the solution field. Examples are the Chimera method[21] or sliding interface approaches.[22-24] It is also worthwhile to mention the immersed boundary method,[25] where simulations are always performed on a cartesian background grid.

Deforming domain problems are inherently transient and their solution requires a form of time discretization. Common approaches include time-stepping methods which separate spatial and temporal discretization and space-time methods which apply a combined discretization to the space-time domain. Time stepping schemes, for example, the generalized α-method,[26] require an arbitrary Lagrangian–Eulerian (ALE) formulation for moving-domain simulations,[27,28] while the space-time formulation directly accounts for the (spatial) mesh deformation.[29,30]

If the movement is known during mesh generation—typically this means before the simulation start—the time-continuous space-time approach allows to incorporate complex deformations of the spatial domain in a boundary-conforming space-time mesh. Even topology changes can be included as shown in finite volume and finite element simulations.[31,32] For two-dimensional problems, standard mesh generation tools can be used to construct the three-dimensional space-time mesh. For three-dimensional problems, more sophisticated mesh generation and adaptation techniques are required. The common approach to generate an unstructured four-dimensional mesh is based on the extrusion of a tetrahedral mesh followed by a subdivision of the prismatic elements into pentatopes (four-simplices). The subdivision can either be achieved with an element-wise Delaunay triangulation[33] or with a predefined decomposition which requires a consistently numbered tetrahedral mesh.[34] Further techniques enable refinement and anisotropic adaptation of four-dimensional meshes.[13,35,36] Four-dimensional meshes with complex deformations and topology changes of the three-dimensional spatial domain can be obtained with an elastic mesh update following extrusion-based pentatope mesh generation.[37] Please, note that the additional effort for the space-time mesh generation completely replaces the special treatment of a deforming domain problem during the simulation. Nevertheless, our boundary-conforming space-time mesh approach is limited to 4D geometries that can be obtained by extrusion of a 3D geometry and a subsequent elastic deformation. Generating meshes for general 4D geometries of engineering scale is—to the best of our knowledge—an open research problem.

3.2 | Discrete formulation for the full-order model

Next, we will derive the FOM, which will be based on the FEM. Thus, we introduce corresponding finite-dimensional subspaces for the trial and weighting spaces introduced in Section 2. Let $\mathcal{S}_{\boldsymbol{u}}^{h}(Q)$ and $\mathcal{V}_{\boldsymbol{u}}^{h}(Q)$ be the finite-dimensional subspaces of $\mathcal{S}_{\boldsymbol{u}}(Q)$ and $\mathcal{V}_{\boldsymbol{u}}(Q)$, respectively. Furthermore, $\mathcal{S}_p^h(Q)$ is the finite-dimensional subspace of $\mathcal{S}_p(Q)$. We stick to the *time-continuous simplex space-time (C-SST)* approach, that is, the computational mesh will be composed of simplex elements filling the entire space-time domain Q. Furthermore, we use first-order polynomials as shape functions $\boldsymbol{N}_i^{\boldsymbol{u}}$ and N_i^p for velocity and pressure, respectively.

To handle (parametric) Dirichlet boundary conditions, we introduce a *velocity lifting function* $\boldsymbol{l}^h(\boldsymbol{\mu})$ such that the *discrete velocity trial function* is given as

$$\boldsymbol{u}^h(\boldsymbol{\mu}) = \boldsymbol{v}^h(\boldsymbol{\mu}) + \boldsymbol{l}^h(\boldsymbol{\mu}), \tag{12}$$

with the homogeneous portion $\boldsymbol{v}^h(\boldsymbol{\mu})$ and $\boldsymbol{l}^h(\boldsymbol{\mu})|_{P_\mathrm{D}} = \boldsymbol{u}_\mathrm{D}(\boldsymbol{\mu})$. Note that additional lifting functions may be used to separate parameter-dependent and independent portions of the Dirichlet boundary conditions. For the sake of notation, however, we only consider the case of a single lifting function here. Consequently, it holds that $\boldsymbol{u}^h \in \mathcal{S}_{\boldsymbol{u}}^h(Q)$ and $\boldsymbol{v}^h \in \mathcal{V}_{\boldsymbol{u}}^h(Q)$. Moreover, the *discrete pressure trial function* is denoted as $p^h(\boldsymbol{\mu}) \in \mathcal{S}_p^h(Q)$.

We apply the Galerkin-least-squares (GLS) stabilization technique,[38-41] which adds a least-squares form of the residual within each element to the original variational formulation of the problem. In the C-SST formulation, this will apply to space-time elements denoted by Q^e.

Following the description, for example, from here, References 42,43 the resulting space-time Galerkin formulation including the additional stabilization terms reads:

Find $\left(\boldsymbol{v}^h(\boldsymbol{\mu}), p^h(\boldsymbol{\mu})\right) \in \mathcal{V}_{\boldsymbol{u}}^h(Q) \times \mathcal{S}_p^h(Q)$, *such that* $\forall \left(\boldsymbol{w}^h, q^h\right) \in \mathcal{V}_{\boldsymbol{u}}^h(Q) \times \mathcal{S}_p^h(Q)$:

$$\begin{aligned}\left(\boldsymbol{w}^h, \boldsymbol{v}_t^h; \boldsymbol{\mu}\right)_Q - b\left(\boldsymbol{w}^h, p^h; \boldsymbol{\mu}\right)_Q + a\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q + b\left(\boldsymbol{v}^h, q^h; \boldsymbol{\mu}\right)_Q + s_\mathrm{MOM}\left(q^h, \boldsymbol{u}^h, p^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q \\ = \left(\boldsymbol{w}^h, \boldsymbol{f}\right)_Q + \left(\boldsymbol{w}^h, \boldsymbol{h}; \boldsymbol{\mu}\right)_{P_\mathrm{N}} - \left(\boldsymbol{w}^h, \boldsymbol{l}_t^h; \boldsymbol{\mu}\right)_Q - a\left(\boldsymbol{w}^h, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q - b\left(\boldsymbol{l}^h, q^h; \boldsymbol{\mu}\right)_Q,\end{aligned} \tag{13}$$

with

$$s_\mathrm{MOM}\left(q^h, \boldsymbol{u}^h, p^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q = s^{\boldsymbol{v}}\left(q^h, \boldsymbol{v}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q + s^{\boldsymbol{l}}\left(q^h, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q + s^p\left(q^h, p^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q, \tag{14}$$

and

$$s^{\boldsymbol{v}}\left(q^h, \boldsymbol{v}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q = \sum_e \int_{Q^e} \tau_\mathrm{MOM}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \frac{1}{\rho}\left(-\boldsymbol{\nabla} q^h\right) \cdot \left(\rho \boldsymbol{v}_t^h(\boldsymbol{\mu})\right) dQ, \tag{15}$$

$$s^{\boldsymbol{l}}\left(q^h, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q = \sum_e \int_{Q^e} \tau_\mathrm{MOM}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \frac{1}{\rho}\left(-\boldsymbol{\nabla} q^h\right) \cdot \left(\rho \boldsymbol{l}_t^h(\boldsymbol{\mu})\right) dQ, \tag{16}$$

$$s^p\left(q^h, p^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q = \sum_e \int_{Q^e} \tau_\mathrm{MOM}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \frac{1}{\rho} \boldsymbol{\nabla} q^h \cdot \boldsymbol{\nabla} p^h(\boldsymbol{\mu})\, dQ. \tag{17}$$

Note that the subscripts of s_MOM and τ_MOM refer to the connection of these terms to the momentum equation stated in Equation (3). The stabilization parameter $\tau_\mathrm{MOM}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right)$ is chosen as previously presented.[44] Although the formulation in detail is not of great importance for this work, note that it depends both on the parametric velocity $\boldsymbol{u}^h(\boldsymbol{\mu})$ and the parametric viscosity $\eta(\boldsymbol{u}^h, \boldsymbol{\mu})$ in a non-linear way. Furthermore, the second-order derivatives of the velocity weighting and trial functions, which appear in the original formulation of the momentum stabilization, are omitted due to the first-order linear basis functions in use.

As a foundation for the description of the ROM in the following section, we present next the *algebraic formulation* of the problem. The vectors of coefficients are denoted as $\boldsymbol{U} \in \mathbb{R}^{N_{\boldsymbol{u}}^h}$ and $\boldsymbol{P} \in \mathbb{R}^{N_p^h}$ for the homogeneous velocity $\boldsymbol{v}^h$ and pressure field p^h, respectively. Here, $N_{\boldsymbol{u}}^h$ and N_p^h stand for the number of degrees of freedom (DOFs) in the FOM. The solution can then be computed by solving the following non-linear system for $\boldsymbol{U}(\boldsymbol{\mu})$ and $\boldsymbol{P}(\boldsymbol{\mu})$:

$$\begin{bmatrix} \boldsymbol{E} + \boldsymbol{A}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) & -\boldsymbol{B}^\mathrm{T} \\ \boldsymbol{B} + \boldsymbol{C}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) & \boldsymbol{S}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}(\boldsymbol{\mu}) \\ \boldsymbol{P}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{H} + \boldsymbol{F} + \boldsymbol{L}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \\ \boldsymbol{G} + \boldsymbol{D}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \end{bmatrix}, \tag{18}$$

TABLE 1 Dimensions of LHS matrices and RHS vectors for the FOM.

LHS matrices	Dimensions	RHS vectors	Dimensions
$\boldsymbol{E}, \boldsymbol{A}$	$\mathbb{R}^{N_{\boldsymbol{u}}^h \times N_{\boldsymbol{u}}^h}$	$\boldsymbol{H}, \boldsymbol{F}, \boldsymbol{L}$	$\mathbb{R}^{N_{\boldsymbol{u}}^h}$
$\boldsymbol{B}, \boldsymbol{C}$	$\mathbb{R}^{N_p^h \times N_{\boldsymbol{u}}^h}$	$\boldsymbol{G}, \boldsymbol{D}$	$\mathbb{R}^{N_p^h}$
$\boldsymbol{S}$	$\mathbb{R}^{N_p^h \times N_p^h}$		

where the left-hand side (LHS) matrices for $i, j = 1, \ldots, N_{\boldsymbol{u}}^h$ and for $k, l = 1, \ldots, N_p^h$ are given as

$$\boldsymbol{E} = \left[E_{i,j}\right], \text{ with } E_{i,j} = \left(\boldsymbol{N}_i^{\boldsymbol{u}}, \frac{\partial \boldsymbol{N}_j^{\boldsymbol{u}}}{\partial t}\right)_Q, \qquad \boldsymbol{A} = \left[A_{i,j}\right], \text{ with } A_{i,j} = a\left(\boldsymbol{N}_i^{\boldsymbol{u}}, \boldsymbol{N}_j^{\boldsymbol{u}}; \boldsymbol{u}^h, \boldsymbol{\mu}\right),$$

$$\boldsymbol{B} = \left[B_{k,j}\right], \text{ with } B_{k,j} = b\left(\boldsymbol{N}_j^{\boldsymbol{u}}, N_k^p\right)_Q, \qquad \boldsymbol{C} = \left[C_{k,j}\right], \text{ with } C_{k,j} = s^{\boldsymbol{v}}\left(N_k^p, \boldsymbol{N}_j^{\boldsymbol{u}}; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q,$$

$$\boldsymbol{S} = \left[S_{k,l}\right], \text{ with } S_{k,l} = s^p\left(N_k^p, N_l^p; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q,$$

and the right-hand side (RHS) vectors read

$$\boldsymbol{H} = \{H_i\}, \text{ with } H_i = -\left(\boldsymbol{N}_i^{\boldsymbol{u}}, \boldsymbol{l}_t^h; \boldsymbol{\mu}\right)_Q, \qquad \boldsymbol{F} = \{F_i\}, \text{ with } F_i = \left(\boldsymbol{N}_i^{\boldsymbol{u}}, h; \boldsymbol{\mu}\right)_{P_{\mathrm{N}}},$$

$$\boldsymbol{L} = \{L_i\}, \text{ with } L_i = -a\left(\boldsymbol{N}_i^{\boldsymbol{u}}, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right), \qquad \boldsymbol{G} = \{G_k\}, \text{ with } G_k = -b\left(\boldsymbol{l}^h, N_k^p; \boldsymbol{\mu}\right)_Q,$$

$$\boldsymbol{D} = \{D_k\}, \text{ with } D_k = -s^{\boldsymbol{l}}\left(N_k^p, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q.$$

For convenience, the dimensions of the matrices and vectors are summarized in Table 1.

4 | REDUCED-ORDER MODEL FOR FLUID FLOW IN DEFORMING DOMAINS

Now that the FOM has been defined, we can turn to the MOR. We will start with a description of the underlying ideas of MOR in Section 4.1 before the ROM is eventually constructed via POD with subsequent projection in Section 4.2.

4.1 | Methodological background for the reduced-order model

In the context of MOR for numerical schemes, one can distinguish between *interpolation*- and *projection*-based approaches.[45] Methods of the former type try to directly capture input-output relations on the basis of data that comes from numerical simulations or measurements. The latter project the underlying governing equations onto a reduced space that has been constructed before. The class of projection-based methods further entails *certified Reduced Basis (RB)* methods[46] or *POD-projection* methods.[45-47] It is also worthwhile to mention the *proper generalized decomposition (PGD)* here.[47] The applicability and effectiveness of these methods depend on the class of problems under consideration. For example, in the realm of elliptic and steady problems, an extensive amount of research and scholarly work exists.[48-53] For a more detailed overview, see also these textbooks[46,54] and references therein. In contrast, time-dependent problems pose additional challenges, just as computational domains undergoing deformation. Both aspects are present in the problems considered here. Therefore, we will focus on that subdomain of existing MOR techniques in which the unsteady nature of a problem or a deforming computational domain is addressed.

To account for the transient character of the problem in the reduction process, there exist two alternative ways of handling time.[55] The first one follows the classical time-stepping approach. In this context, RB approximations have been applied to a two-dimensional transient heat conduction problem.[56] Based on so-called POD-greedy methods,[57,58] linear and non-linear problems have been tackled as well. While the governing equations considered range from convection-diffusion problems[57] or Burger's equation and porous media flow[59] to the compressible Navier-Stokes equations,[60] all applications are limited to at most two-dimensional spatial domains. Additional work presents a reduction technique for the long-time integration of incompressible turbulent flows,[61] but also analyzes only a two-dimensional lid-driven cavity flow. The second option for handling time is rather based on space-time formulations, which we consider in this work. They have been used in the context of RB methods to derive effective long-time a-posteriori error bounds for linear[62,63] and non-linear[64,65] parabolic PDEs. Another study presented a space-time least-squares Petrov–Galerkin projection method for the reduction of nonlinear dynamical systems, based on a tensor decomposition of snapshot data and providing error bounds.[66] However, the model problems in the aforementioned works involve only scalar unknowns and up to two spatial dimensions. A staggered RB method in space-time for heat transfer and convection-diffusion problems has also been introduced,[67] but it is limited to two spatial dimensions and structured meshes. The space-time perspective has further been used to tackle, for example, time-dependent optimal control problems.[68,69] In addition, a space-time ROM for sub-intervals of the total simulation time has been presented.[70] The interested reader may also refer to recent works on the state of the art in RB methods for time-dependent problems.[71,72] In distinction to the former methods and for the first time, we apply a POD-projection approach to the time-continuous space-time formulation. It is this specific combination of existing approaches by which we extend the range of available reduction techniques for transient problems. In particular, we are able to tackle problems defined in up to three-dimensional space, resulting in up to four-dimensional space-time domains. The respective computational meshes can be unstructured, consisting of simplex elements. Furthermore, we apply the method to a complex class of problems, that is, a non-linear form of the Stokes equations. It should be noted, however, that due to the generality of this approach, we lack theoretical results such as error estimators. Nevertheless, the presented method provides an alternative to existing reduction methods for time-dependent problems and exhibits its fullest potential not least when applied to problems that additionally involve domain deformations.

For deforming domain problems, the construction of a ROM is usually quite involved and requires some careful treatment of the deformations applied to the computational mesh.[73,74] This is due to the fact that the FEM function spaces are inherently linked with the geometry of the underlying grid. A strategy based on a mapping functional to relate the time-dependent solution in one- and two-dimensional deforming domains to fixed reference domains has been developed.[75] Furthermore, temporally-local eigenfunctions have been used for MOR, too.[76] To this respect, the C-SST approach (cf. Section 3.2) offers an appealing but straightforward alternative, since all deformations—as long as they are prescribed or known a-priori—are already integrated into the computational mesh and, thereby, considered in the original function spaces. As a result, the proposed technique can also work with transient problems in complex and deforming geometries, as stated before, a very demanding scenario for MOR. Thus, it means a notable advancement concerning the application area of existing MOR techniques. Also, it is worthwhile to note that this approach even works in the presence of spatial topology changes without any further adaptions.

Although we are interested in spatially deforming domains, we would like to stress that we do *not* consider the MOR of problems with geometrical parametrization here. The latter would need to be taken into account if we consider cases in which the deformation is not known a-priori, for example, fluid-structure interaction problems. Even though the presented approach only works for problems with prescribed motion, the extension to other scenarios with unknown deformation is possible by allowing a parametrization of the geometry. In this case, one would have to resort to one of the existing methods for parameterized domains[77-84] and combine it with the presented approach.

4.2 | Discrete formulation for the reduced-order model

Before we can construct an efficient ROM whose complexity is independent of the FOM, we will have to address the non-linearity appearing in our problem through the viscosity model as well as through the formulation of the stabilization parameter. For this purpose, we make use of the empirical interpolation method (EIM)[85,86] here and introduce the following approximations for the viscosity and the stabilization parameter:

$$\eta\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \approx \sum_{q=1}^{Q_\eta} c_q^\eta\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) h_q^\eta(\boldsymbol{x}, t), \qquad \tau_{\mathrm{MOM}}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \approx \sum_{q=1}^{Q_\tau} c_q^\tau\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) h_q^\tau(\boldsymbol{x}, t).$$

Note that each of the $Q_\star$ terms consists of parameter-dependent coefficients $c_q^\star\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right)$ and parameter-independent basis functions $h_q^\star(\boldsymbol{x}, t)$, where $\star \in \{\eta, \tau\}$. The basis functions $h_q^\star(\boldsymbol{x}, t)$ are defined in the entire space-time domain Q. Consequently, the viscous term is replaced by

$$a\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q \approx \sum_{q=1}^{Q_\eta} c_q^\eta\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) a_q\left(\boldsymbol{w}^h, \boldsymbol{v}^h\right)_Q, \tag{19}$$

with

$$a_q\left(\boldsymbol{w}^h, \boldsymbol{v}^h\right)_Q = \int_Q \epsilon\left(\boldsymbol{w}^h\right) : 2h_q^{\eta}(\boldsymbol{x}, t)\epsilon\left(\boldsymbol{v}^h\right)\, dQ. \tag{20}$$

Moreover, the stabilization term $s^{\boldsymbol{v}}$ is approximated via

$$s^{\boldsymbol{v}}\left(q^h, \boldsymbol{v}^h; \boldsymbol{u}^h, \boldsymbol{\mu}\right)_Q \approx \sum_{q=1}^{Q_\tau} c_q^{\tau}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) s_q^{\boldsymbol{v}}\left(q^h, \boldsymbol{v}^h\right)_Q, \tag{21}$$

where

$$s_q^{\boldsymbol{v}}\left(q^h, \boldsymbol{v}^h\right)_Q = \sum_e \int_{Q^e} h_q()^{\tau}(\boldsymbol{x}, t)\frac{1}{\rho}\left(-\boldsymbol{\nabla} q^h\right) \cdot \left(\rho \boldsymbol{v}_t^h\right)\, dQ. \tag{22}$$

Analogously, the remaining stabilization terms $s^{\boldsymbol{l}}(q^h, \boldsymbol{l}^h; \boldsymbol{u}^h, \boldsymbol{\mu})_Q$ and $s^p(q^h, p^h; \boldsymbol{u}^h \boldsymbol{\mu})_Q$ can be approximated using parameter-independent terms $s_q^{\boldsymbol{l}}(q^h, \boldsymbol{l}^h)_Q$ and $s_q^p(q^h, p^h)_Q$, respectively. As a consequence, all the matrices and vectors in the algebraic system of the FOM relying on these forms, that is, $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{S}$, $\boldsymbol{L}$, and $\boldsymbol{D}$ are replaced by approximations using $\boldsymbol{A}^q$, $\boldsymbol{C}^q$, $\boldsymbol{S}^q$, $\boldsymbol{L}^q$, and $\boldsymbol{D}^q$ corresponding to the respective terms above. For example, this means $\boldsymbol{A}^q = [A_{i,j}^q]$, with $A_{i,j}^q = a_q(\boldsymbol{N}_i^{\boldsymbol{u}}, \boldsymbol{N}_j^{\boldsymbol{u}})_Q$. For the computation of the coefficients $c_q^{\star}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right)$ and basis functions $h_q^{\star}(\boldsymbol{x}, t)$, we use a standard implementation of the EIM,[46] where the space-time coordinates $(\boldsymbol{x}, t)$ are used instead of the spatial coordinates present in the original algorithm.

To perform the projection step later, a basis spanning the reduced spaces is needed. To that end, we apply the POD using the method of snapshots,[87] that is, solutions of the FOM. In particular, this is done individually for the homogeneous velocity $\boldsymbol{v}^h$ and the pressure p^h leading to the *reduced finite-dimensional function spaces* $\mathcal{V}_{\boldsymbol{u}}^N \subset \mathcal{V}_{\boldsymbol{u}}^h$ and $\mathcal{S}_p^N \subset \mathcal{S}_p^h$. As a result of the POD, we obtain $N_{\boldsymbol{v}}$ and N_p basis functions for the reduced representation $(\boldsymbol{v}^N, p^N)$ of the homogeneous velocity and pressure field, respectively. To account for the Dirichlet boundary conditions, the basis for $\mathcal{V}_{\boldsymbol{u}}^N$ is augmented with the lifting function(s) $\boldsymbol{l}^h$ yielding the reduced space $\boldsymbol{\mathcal{S}}_{\boldsymbol{u}}^N \subset \boldsymbol{\mathcal{S}}_{\boldsymbol{u}}^h$ for the reduced velocity $\boldsymbol{u}^N$. Therefore, we use $N_{\boldsymbol{u}} \geq N_{\boldsymbol{v}}$ to denote the number of basis functions for $\boldsymbol{u}^N$. We sort the basis functions in descending order of significance—indicated by the magnitude of the corresponding eigenvalues—while the lifting functions are always leading to ensure that the Dirichlet boundary conditions are met, even if we only use a subset of these basis functions.

For all basis functions, we collect the coefficients with respect to the FOM function spaces in the so-called *basis function matrices* $\boldsymbol{Z}_{\boldsymbol{u}} \in \mathbb{R}^{N_{\boldsymbol{u}}^h \times N_{\boldsymbol{u}}}$ and $\boldsymbol{Z}_p \in \mathbb{R}^{N_p^h \times N_p}$. These matrices are multiplied with those from the algebraic system of the FOM given in Equation (18), which yields the projection of the corresponding operators onto the reduced space. Thus, one can formulate the algebraic system for the vectors of unknowns $\boldsymbol{U}_N \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{P}_N \in \mathbb{R}^{N_p}$, where $N_{\boldsymbol{u}}$ and N_p are the number of unknowns of the ROM. The key assumption for an effective reduction is that it holds that $N = N_{\boldsymbol{u}} + N_p \ll N^h = N_{\boldsymbol{u}}^h + N_p^h$. The system finally reads:

$$\begin{aligned}\begin{bmatrix} \boldsymbol{E}_N + \boldsymbol{A}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) & -\boldsymbol{B}_N^{\mathrm{T}} \\ \boldsymbol{B}_N + \boldsymbol{C}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) & \boldsymbol{S}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_N(\boldsymbol{\mu}) \\ \boldsymbol{P}_N(\boldsymbol{\mu}) \end{bmatrix} \\ = \begin{bmatrix} \boldsymbol{H}_N + \boldsymbol{F}_N + \boldsymbol{L}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \\ \boldsymbol{G}_N + \boldsymbol{D}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \end{bmatrix},\end{aligned} \tag{23}$$

where the LHS matrices read

$$\begin{aligned} &\boldsymbol{E}_N = \boldsymbol{Z}_{\boldsymbol{u}}^{\mathrm{T}} \boldsymbol{E} \boldsymbol{Z}_{\boldsymbol{u}}, \quad && \boldsymbol{A}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) = \sum_{q=1}^{Q_\eta} c_q^{\eta}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \boldsymbol{A}_N^q, \quad \text{and} \quad \boldsymbol{A}_N^q = \boldsymbol{Z}_{\boldsymbol{u}}^{\mathrm{T}} \boldsymbol{A}^q \boldsymbol{Z}_{\boldsymbol{u}}, \\ &\boldsymbol{B}_N = \boldsymbol{Z}_p^{\mathrm{T}} \boldsymbol{B} \boldsymbol{Z}_{\boldsymbol{u}}, && \\ & && \boldsymbol{C}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) = \sum_{q=1}^{Q_\tau} c_q^{\tau}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \boldsymbol{C}_N^q, \quad \text{and} \quad \boldsymbol{C}_N^q = \boldsymbol{Z}_p^{\mathrm{T}} \boldsymbol{C}^q \boldsymbol{Z}_{\boldsymbol{u}}, \\ & && \boldsymbol{S}_N\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) = \sum_{q=1}^{Q_\tau} c_q^{\tau}\left(\boldsymbol{u}^h, \boldsymbol{\mu}\right) \boldsymbol{S}_N^q, \quad \text{and} \quad \boldsymbol{S}_N^q = \boldsymbol{Z}_p^{\mathrm{T}} \boldsymbol{S}^q \boldsymbol{Z}_p, \end{aligned}$$

TABLE 2 Dimensions of LHS matrices and RHS vectors for the ROM.

LHS matrices	Dimensions	RHS vectors	Dimensions
$\boldsymbol{E}_N, \boldsymbol{A}_N$	$\mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$	$\boldsymbol{H}_N, \boldsymbol{F}_N, \boldsymbol{L}_N$	$\mathbb{R}^{N_{\boldsymbol{u}}}$
$\boldsymbol{B}_N, \boldsymbol{C}_N$	$\mathbb{R}^{N_p \times N_{\boldsymbol{u}}}$	$\boldsymbol{G}_N, \boldsymbol{D}_N$	$\mathbb{R}^{N_p}$
$\boldsymbol{S}_N$	$\mathbb{R}^{N_p \times N_p}$		

and the RHS vectors are given as

$$\begin{aligned} \boldsymbol{H}_N &= \boldsymbol{Z}_{\boldsymbol{u}}^{\mathrm{T}}\boldsymbol{H}, \quad & \boldsymbol{L}_N(\boldsymbol{u}^h, \boldsymbol{\mu}) &= \sum_{q=1}^{Q_\eta} c_q^{\eta}(\boldsymbol{u}^h, \boldsymbol{\mu})\boldsymbol{L}_N^q, \quad \text{and} \quad \boldsymbol{L}_N^q = \boldsymbol{Z}_{\boldsymbol{u}}^{\mathrm{T}}\boldsymbol{L}^q, \\ \boldsymbol{F}_N &= \boldsymbol{Z}_{\boldsymbol{u}}^{\mathrm{T}}\boldsymbol{F}, & & \\ \boldsymbol{G}_N &= \boldsymbol{Z}_p^{\mathrm{T}}\boldsymbol{G}, \quad & \boldsymbol{D}_N(\boldsymbol{u}^h, \boldsymbol{\mu}) &= \sum_{q=1}^{Q_\tau} c_q^{\tau}(\boldsymbol{u}^h, \boldsymbol{\mu})\boldsymbol{D}_N^q, \quad \text{and} \quad \boldsymbol{D}_N^q = \boldsymbol{Z}_p^{\mathrm{T}}\boldsymbol{D}^q. \end{aligned}$$

Table 2 gives an overview over the dimensions of the matrices and vectors for the ROM. Due to the reduced dimensions, the solution of the algebraic system of the ROM for any new parameter sample $\boldsymbol{\mu}$ is possible with significantly less computational resources. Thanks to the EIM, the same holds for the assembly process of this system. Note, however, that due to the specific implementation of the full-order solver, we keep the dependency on $\boldsymbol{u}^h$ in the EIM coefficients.

5 | NUMERICAL RESULTS

In the following, we will illustrate how the proposed ROM approach, which makes use of a time-continuous space-time setting, can help to significantly decrease the required computational resources for time-dependent parametric problems defined in deforming domains. To that end, we present error and performance analysis results for two test cases. The test cases are representative of applications that may arise from engineering or biomedical problems, for example. The first test case involves a two-dimensional valve-like geometry, resulting in a 3D space-time geometry. In space, the valve plug moves over time and even closes off parts of the geometry, which means that the spatial topology changes. In the second test case, we consider a three-dimensional artery-like geometry. Consequently, the resulting space-time domain is 4D. In the center region, the geometry is compressed over time, which yields a deforming domain problem.

5.1 | Valve-like geometry with topology changes

In this section, we consider a two-dimensional valve-like geometry composed of a valve plug encased in a flow channel. Its initial configuration for $t = 0\,\mathrm{s}$ is depicted in Figure 2A. At the top, fluid can enter the channel through an inlet with width $L_{\mathrm{inlet}} = 0.025\,\mathrm{m}$. The outlet is located at the bottom. All remaining boundary portions are impermeable. We consider a time interval such that $t \in [0, 1.8]\,\mathrm{s}$. With the passage of time, the plug starts to move inward for $t \in [0.3, 0.7)\,\mathrm{s}$, opening a second branch for the flow on the left-hand side. After reaching the center of the casing, the plug stays at rest for $t \in [0.7, 1.1)\,\mathrm{s}$ and, still a bit later, the movement is reversed for $t \in [1.1, 1.5)\,\mathrm{s}$ to close the emerged branch for the remainder of the simulation. The resulting three-dimensional space-time domain is shown in Figure 2B and the computational mesh can be produced with standard meshing tools. The spatial geometry for different time instances can be seen in Figure 3. The velocity of the plug in x-direction is $u_{\mathrm{plug}} = \pm 0.0625\ \mathrm{m\ s^{-1}}$ for the inward and outward movement, respectively.

The fluid is supposed to be plastics melt as present, for example, in various polymer processing techniques for thermoplastics. We use material properties for an exemplary polycarbonate (PC) offered by Covestro Makrolon®. We use the density $\rho = 1200\ \mathrm{kg\ m^{-3}}$ and, following existing literature,[88] the parameters for the viscosity model from Equation (9) as

$$\eta_0 = 270\ \mathrm{Pa\ s}, \ \eta_\infty = 0\mathrm{Pa\ s}, \ \lambda = 1.2 \times 10^{-3}\ \mathrm{s}, \ a = 1, \ n = 0.775. \tag{24}$$

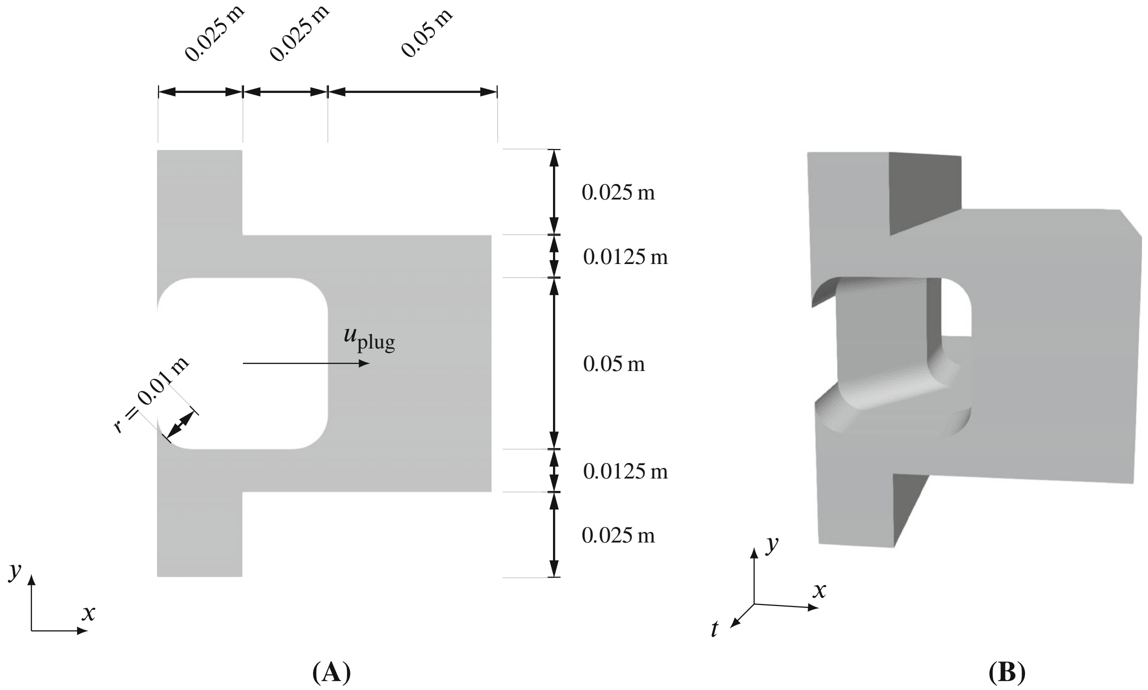


FIGURE 2 Computational domain of valve-like test case. (A) Spatial domain for $t = 0.0$ s. (B) Space-time domain.

In the following, the velocity vector $\boldsymbol{u} = [u, v]^T$ will collect the velocity components u and v for the x- and y-direction, respectively. We prescribe a time-dependent inflow profile given as

$$u_{\text{in}}(x,t) = 0 \text{ m s}^{-1},$$
$$v_{\text{in}}(x,t) = v_{\text{in}}^0(x(x - 0.025 \text{ m}))\sqrt{\frac{t}{1.8 \text{ s}}}.$$

Choosing $v_{\text{in}}^0 = 640 \text{ m}^{-1}$ s, this leads to a Reynolds number of approximately $Re = 1 \times 10^{-2}$ such that the Stokes equations are appropriate to model this creeping flow. For the walls of the casing, no-slip boundary conditions are assumed. At the outlet, a parallel outflow is enforced by setting the velocity in x-direction to zero, that is, $u_{\text{out}} = 0 \text{ m s}^{-1}$. The resulting flow field is visualized in Figure 3 for several points in time.

5.1.1 | ROM for the flow of plastics melt in a valve-like geometry

In the following, we will consider uncertainties in the model parameters from the viscosity model stated in Equation (9). Therefore, we collect two of those parameters in the parameter vector, that is, $\boldsymbol{\mu} = [\lambda, n]$. Assuming that variations of ±5% may occur, it follows that $\boldsymbol{\mu} \in \left[0.95\overline{\lambda}, 1.05\overline{\lambda}\right] \times \left[0.95\overline{n}, 1.05\overline{n}\right]$, with $\overline{\lambda} = 1.2 \times 10^{-3}$ s and $\overline{n} = 0.775$.

Next, we will construct the ROM. To create the snapshots, we generate $N_{\text{train}} = 256$ training samples that are equidistantly spaced over the parameter domain. Note that since we rely on POD to construct the basis, the number of necessary snapshots to compute is not known a-priori or determined dynamically. Thus, we choose N_{train} conservatively, that is, large enough to capture all relevant information in the snapshots. These snapshots are used both for the POD and the EIM. For an analysis of how the ROM is affected by different numbers of training samples, see Section A.1.

Applying the POD results in the distribution of eigenvalues λ_i shown in Figure 4A. Based on a threshold for the so-called retained energy, we choose $N_{\boldsymbol{v}} = 1, \dots, 10$ and $N_p = 1, \dots, 6$ in the following. Since all Dirichlet boundary conditions are parameter-independent in this case, we use only one lifting function resulting in $N_{\boldsymbol{u}} = 1, \dots, 11$. For comparison, the FOM includes a total number of DOFs of $N^h = 334{,}762$. As has been mentioned before, the non-linearity in the viscosity η and in the stabilization parameter τ_{MOM} is tackled by the EIM. The maximum interpolation error for both

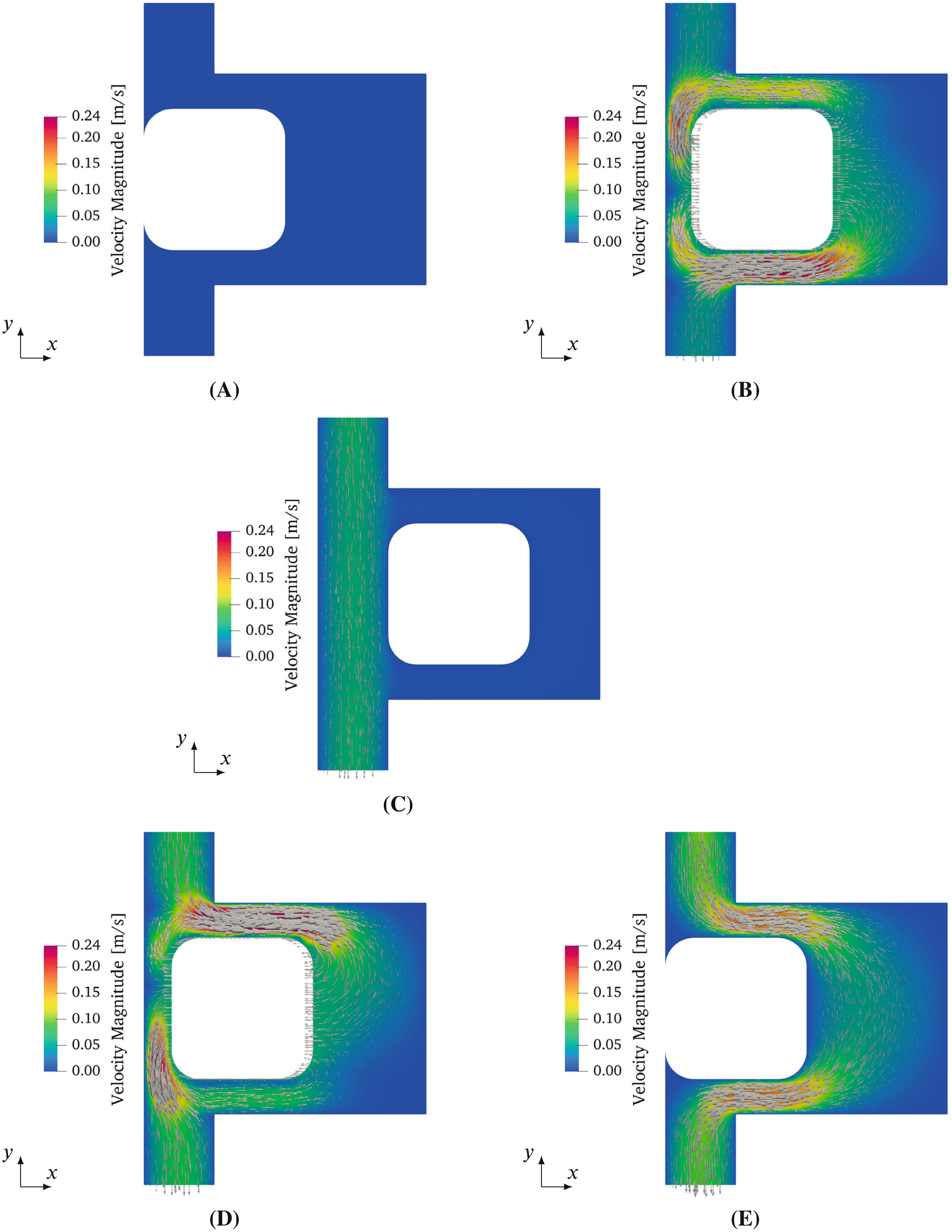


FIGURE 3 Valve-like test case: Flow field for different points in time. (A) $t = 0.0$ s. (B) $t = 0.45$ s. (C) $t = 0.9$ s. (D) $t = 1.35$ s. (E) $t = 1.8$ s.

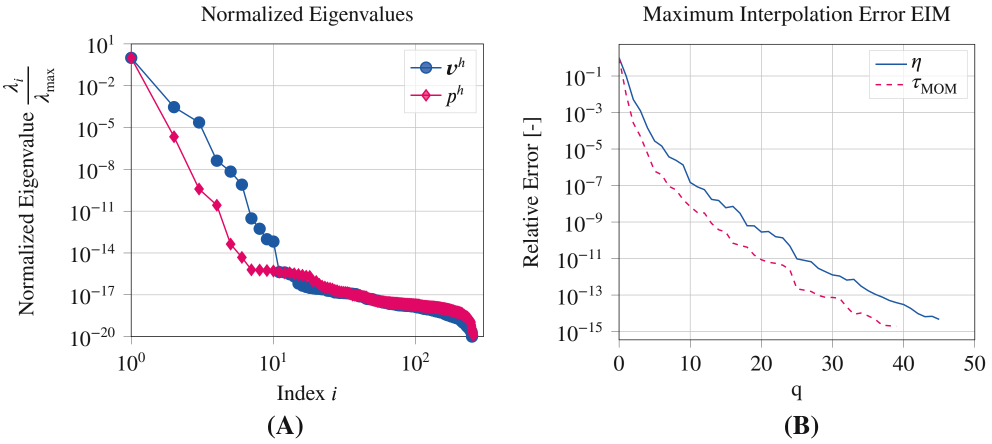


FIGURE 4 Valve-like test case: Results from the construction of the ROM. (A) Distribution of the eigenvalues from the POD. (B) Maximum interpolation error during the EIM.

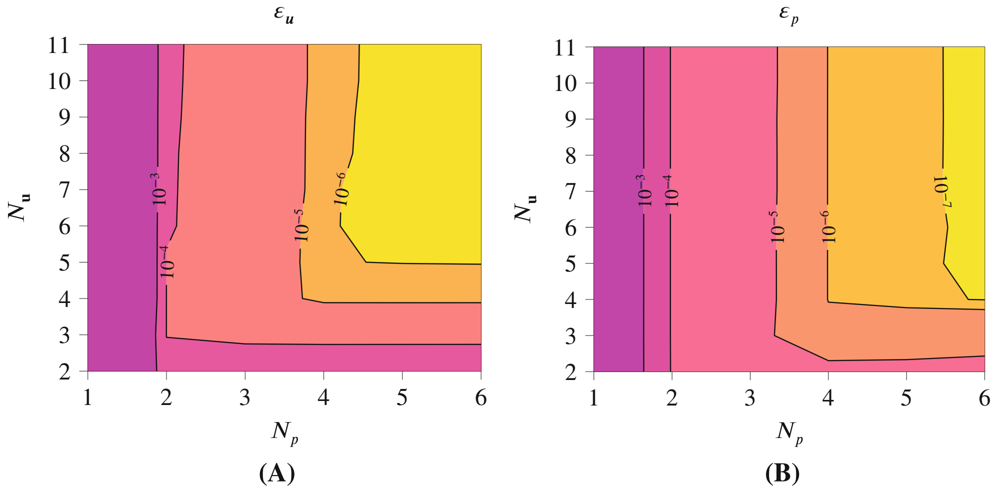


FIGURE 5 Valve-like test case: Maximum relative error of the ROM over all testing samples. (A) Velocity. (B) Pressure.

quantities in the course of the greedy search is depicted in Figure 4B. Setting a tolerance of 5×10^{-15} and 2×10^{-15}, we use $Q_\eta = 45$ and $Q_\tau = 39$ terms for the approximation.

To quantify the quality of the ROM, we perform an error and performance analysis. To that end, we create $N_{\text{test}} = 50$ testing samples drawn from a uniform distribution. For each sample, we compare the solutions from the FOM and the ROM as well as the respective runtimes. To evaluate the accuracy of the ROM, we use the following error definitions:

$$\varepsilon_{\boldsymbol{u}} = \frac{|\boldsymbol{u}^h - \boldsymbol{u}^N|_{\boldsymbol{\mathcal{H}}^1}}{|\boldsymbol{u}^h|_{\boldsymbol{\mathcal{H}}^1}}, \quad \varepsilon_p = \frac{||p^h - p^N||_{L^2}}{||p^h||_{L^2}}, \tag{25}$$

where $|\cdot|_{\boldsymbol{\mathcal{H}}^1}$ and $||\cdot||_{L^2}$ are discrete measures for the $\boldsymbol{\mathcal{H}}^1$ semi-norm and the L^2 norm, respectively. The maximal error over all testing samples is shown in Figure 5, using different numbers of basis functions for the velocity and the pressure field. Note that we ignore the lifting function in these plots, since it does not represent a solution, but is only intended to ensure the Dirichlet boundary conditions. For the velocity, the error ranges from values smaller than 1×10^{-2} to values smaller than 1×10^{-6} when increasing the number of basis functions. Similarly, the maximum error in the pressure is limited by 1×10^{-2} and drops below 1×10^{-7} for $N_{\boldsymbol{u}}$ and N_p large enough. These results suggest that the error introduced by the ROM is in a reasonable range for typical engineering applications and, furthermore, that it can be controlled by choosing the number of basis functions such that a desired accuracy is achieved.

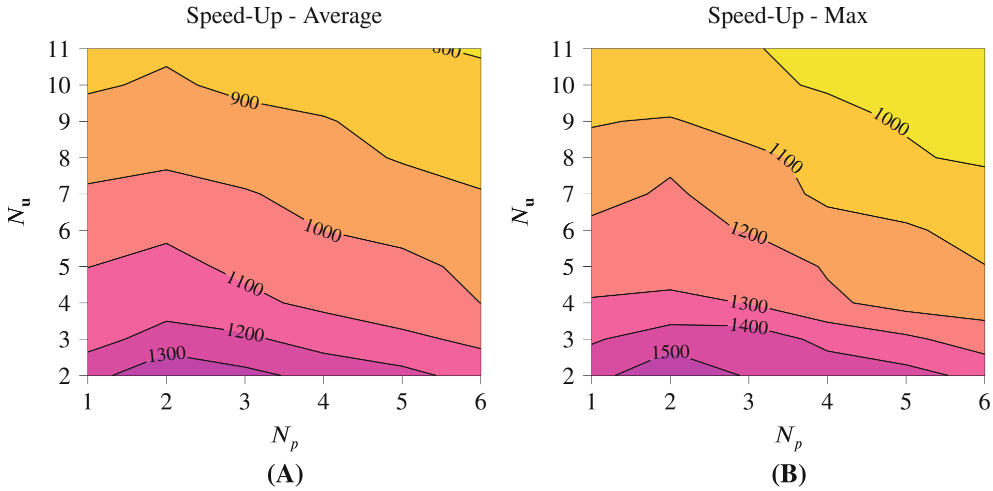


FIGURE 6 Valve-like test case: Performance results. (A) Average speed-up over all testing samples. (B) Maximum speed-up over all testing samples.

TABLE 3 Valve-like test case: Values of the runtime of the FOM over the testing set.

	Minimum	Median	Maximum
$T_{\mathrm{CPU}}^{\mathrm{FOM}}$ [s]	831.59	873.37	982.11

Finally, we investigate the performance of the ROM by comparing the CPU time needed for a single evaluation of the ROM to that for the FOM. Note that the former has been run on a single core, whereas the latter used 64 cores. Also in this analysis, we skip the lifting function for the same reason as before. Figure 6 presents the speed-up, that is, the ratio between the CPU time of FOM and ROM evaluations, for different numbers of basis functions $N_{\boldsymbol{u}}$ and N_p. Both, the average and the maximum speed-up, are in the order of 1000 and indicate that a significant reduction of the runtime for the ROM is realized. In order to be able to calculate the speed-up also with respect to another FOM of choice, for example, a semi-discrete one, we also report the absolute CPU times of our FOM denoted by $T_{\mathrm{CPU}}^{\mathrm{FOM}}$. Together with the runtime of the method to be compared, the speed-up results presented here can be converted. Table 3 states the minimum, median, and maximum values of $T_{\mathrm{CPU}}^{\mathrm{FOM}}$ over the testing set.

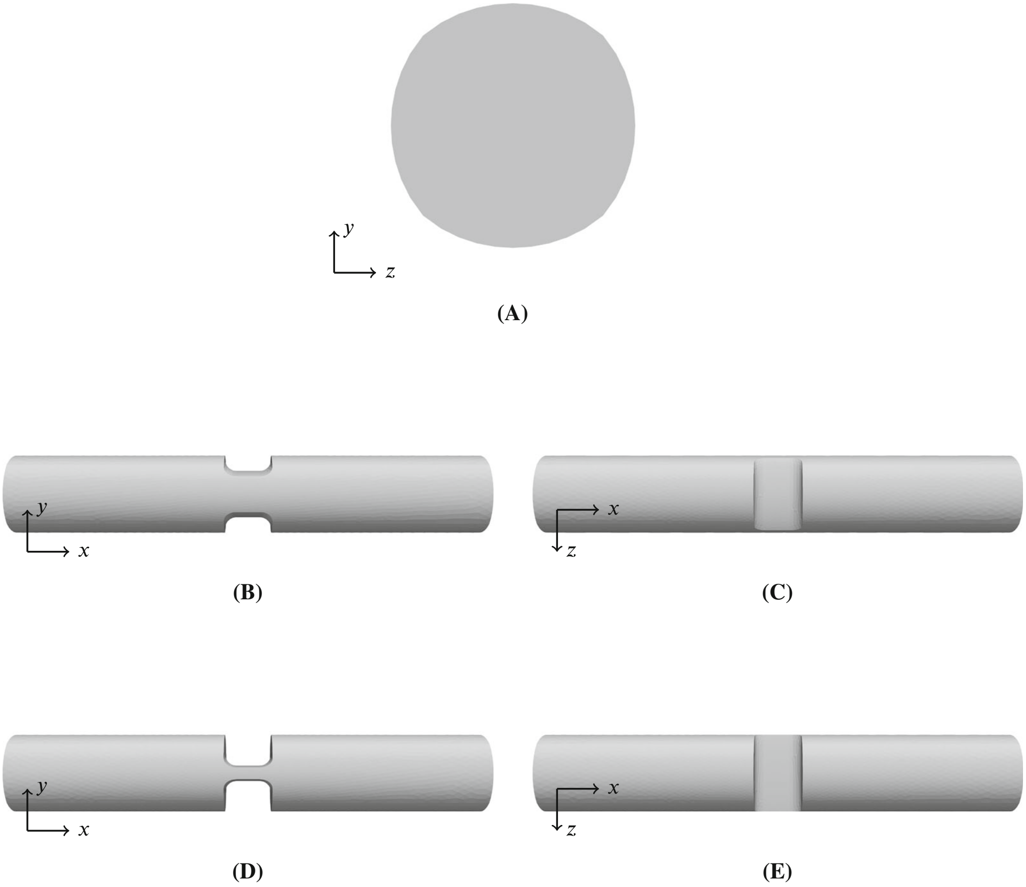
Taken together, the error and performance analysis confirms the effectiveness of the approach for this two-dimensional deforming domain problem with topology changes. In particular, the accuracy of the ROM is acceptable as well as controllable while a significant reduction of the computational demands is achieved, too. This qualifies the ROM as a surrogate model, for example, in one of the aforementioned many query scenarios.

5.2 | Artery-like geometry with compression

After presenting results for a spatially deforming two-dimensional geometry, we will now demonstrate the aptitude of the proposed approach also for the three-dimensional case, resulting in a four-dimensional space-time domain. The geometry is inspired by an artery that locally undergoes compression over time. The initial spatial geometry can be seen in Figure 7A–C. It has a length of $L = 60 \times 10^{-3}$ m and a radius of $r_0 = 5 \times 10^{-3}$ m. We are interested in the internal flow over a time period of 1 s where fluid enters on the left-hand side, that is, at $x_{\min}$. The local narrowing of the artery happens according to the following expression for the upper and lower parts of the moving boundary:

$$y(t) = \pm\left[0.2 + 0.2 \cdot \left(\cos(\pi\ t\ \mathrm{s}^{-1}) + 1\right)\right] \cdot r_0.$$

The final state of the deformed spatial geometry is depicted in Figure 7D,E.



(A)

(B)

(C)

(D)

(E)

FIGURE 7 Artery-like test case: Geometry. (A) Front view. (B) Side view at $t = 0.0$ s. (C) Top view at $t = 0.0$ s. (D) Side view at $t = 1.0$ s. (E) Top view at $t = 1.0$ s.

To mimic blood, the density is set to $\rho = 1058$ kg m^{-3} and the parameters for the viscosity model are chosen as presented in Reference 89:

$$\eta_0 = 0.056 \text{ Pa} \cdot \text{s}, \ \eta_\infty = 0.00345 \text{ Pa} \cdot \text{s}, \ \lambda = 1.902 \text{ s}, \ a = 1.25, \ n = 0.22.$$

For the inflow velocity, we prescribe the following time-dependent profile:

$$u = u_{\text{in}}^0 \left(1 - \frac{(y^2+z^2)}{r_0^2}\right) \cdot \begin{cases} \sqrt{\frac{t}{0.2 \text{ s}}} & \text{for } t < 0.2 \text{ s}, \\ 1 & \text{for } t \geq 0.2 \text{ s}, \end{cases} \quad v = 0 \text{ m s}^{-1}, \quad w = 0 \text{ m s}^{-1},$$

with the velocity vector $\boldsymbol{u} = [u, v, w]^T$ and $u_{\text{in}}^0 = 0.1$ m s^{-1}. At the outlet, a parallel outflow is enforced, that is, $v = w = 0$. Along the walls, no-slip conditions are set. To account for the narrowing, we apply the following boundary conditions for the velocity in y-direction on the horizontal and rounded parts:

$$v = \frac{\partial y(t)}{\partial t} = \mp \pi \sin(\pi t) \times 10^{-3} \ \text{ m s}^{-1}. \tag{26}$$

The sign of this term depends on whether you consider the upper or the lower part. In particular, the negative and the positive sign correspond to a downward movement for $y > 0$ and an upward movement for $y < 0$, respectively. Figure 8 shows the resulting velocity field along the artery in its center plane for the initial and final state.

For this geometry, a locally refined boundary-conforming simplex space-time mesh is constructed through extrusion of a three-dimensional mesh[33] and using the four-dimensional elastic mesh update method (4DEMUM).[37] In particular, we start with a tetrahedral mesh with extension in the x-, y-, and t-directions, as shown in Figure 9. It can be

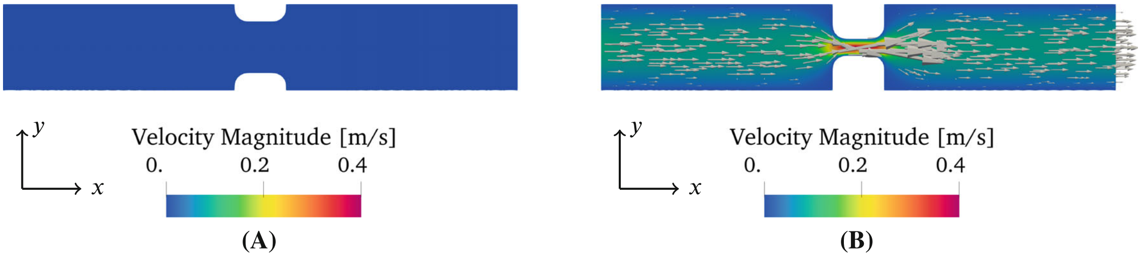


FIGURE 8 Artery-like test case: Velocity field. (A) Initial state at $t = 0.0$ s. (B) Final state at $t = 1.0$ s.

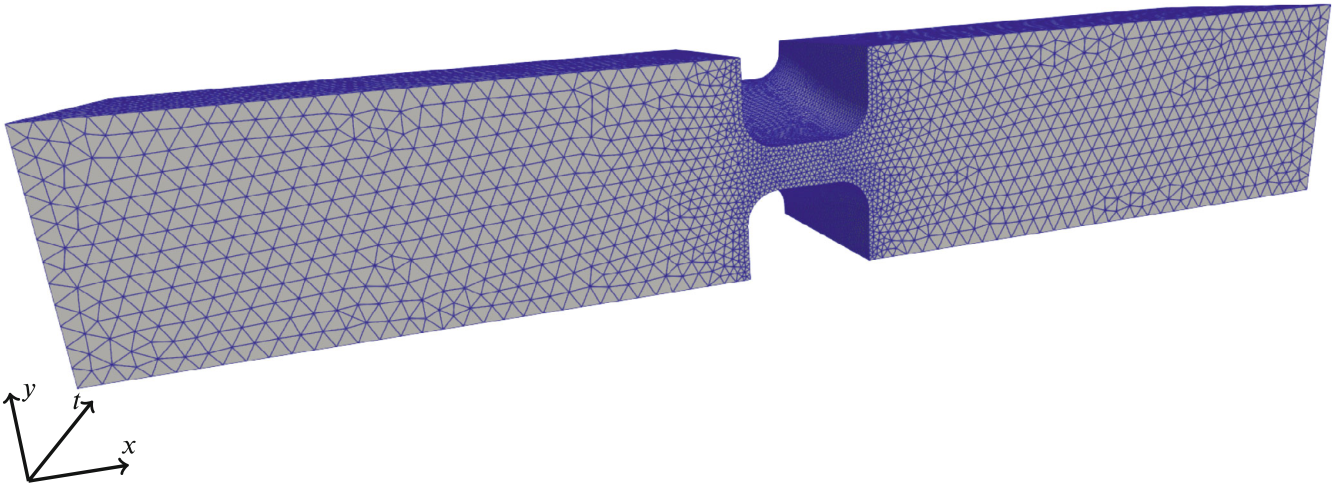


FIGURE 9 Artery-like test case: Three-dimensional x-y-t-mesh with local refinement.

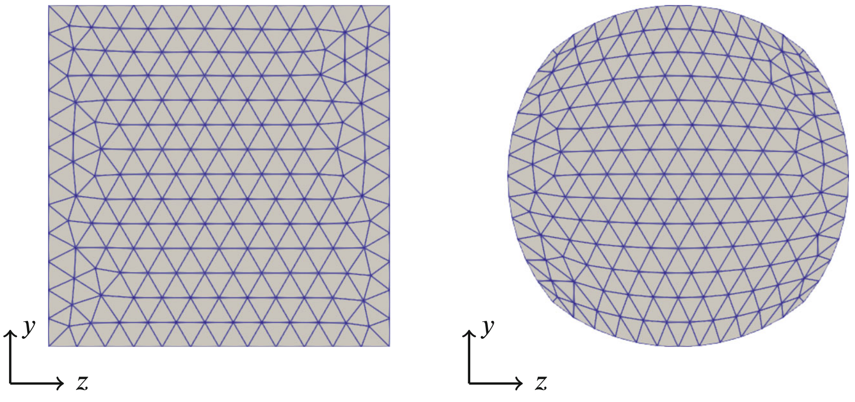
thought of as a space-time grid for the two-dimensional center plane ($z = 0$). Note that in this mesh we can refine elements in the deforming region. Then, the mesh is extruded in the fourth dimension, which inserts nodes along the z-direction and generates a pentatope triangulation of the four-dimensional space-time domain. After extrusion, however, the mesh cross-section at $x = 0$, $t = 0$ is a square. To obtain the approximately circular cross-section of the artery-like geometry, we apply the aforementioned 4DEMUM. A representative example is visualized in Figure 10.

5.2.1 | ROM for the flow of blood in an artery-like geometry

In the following, a ROM is constructed for a variation of the prescribed inflow velocity, that is, $\boldsymbol{\mu} = [u^0_{\text{in}}]$ with $\boldsymbol{\mu} \in [0.95u^0_{\text{in}}, 1.05u^0_{\text{in}}]$. In this case, we have to distinguish between the inlet boundary portion with a parameter-dependent Dirichlet boundary condition and the moving artery walls with prescribed values that are non-zero but parameter-independent. Thus, we make use of two lifting functions here.

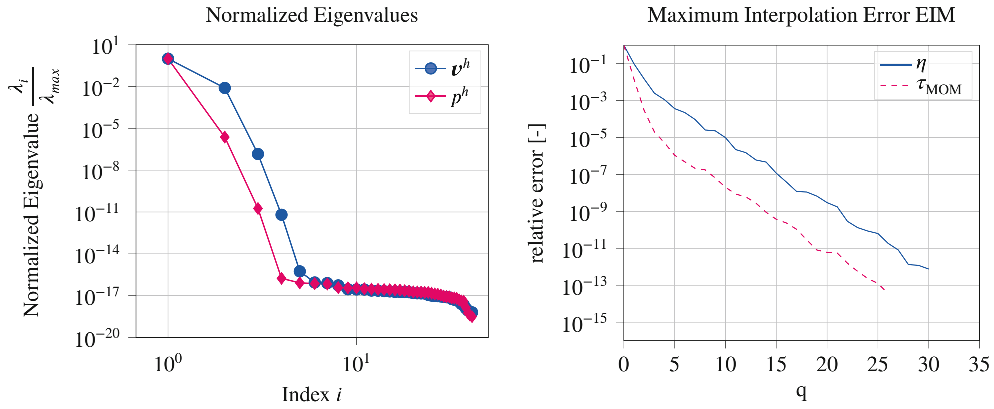
First, we compute snapshots with the FOM for $N_{\text{train}} = 41$ training samples that are equidistantly distributed. Refer to Section A.2 to see how the ROM is affected by different numbers of training samples. Here, the FOM involves $N^h = 2{,}194{,}390$ DOFs in total. As a result of the POD, the distribution of eigenvalues is depicted in Figure 11A and we choose $N_{\boldsymbol{u}} = 1, \ldots, 6$, including the two velocity lifting functions, and $N_p = 1, \ldots, 3$. For the EIM, we set a tolerance of 1×10^{-12} and 1×10^{-13} and obtain $Q_\eta = 30$ and $Q_\tau = 26$, respectively. Figure 11B shows the maximum interpolation error for the viscosity η and the stabilization parameter τ_{MOM} during the EIM.

Subsequently, we use $N_{\text{test}} = 20$ uniformly distributed random samples to carry out the error and performance analysis. The results for the maximal relative errors $\varepsilon_{\boldsymbol{u}}$ and ε_p are presented in Figure 12. The qualitative behavior of the errors is very similar as for the valve-like test case (see Section 5.1.1). For the velocity, it is smaller than 1×10^{-2} and decreases to below 1×10^{-6}. In a like manner, the error in the pressure ranges from values smaller than 1×10^{-2} to values less than 1×10^{-5}.



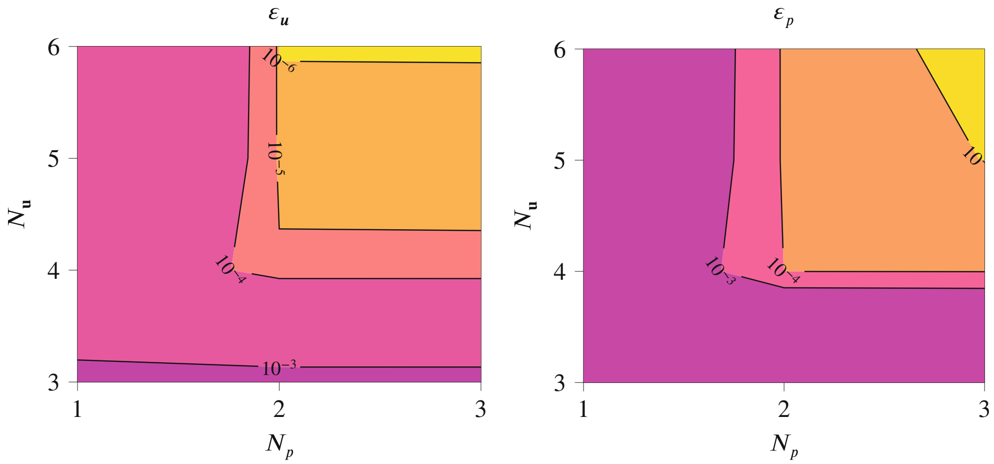
(A) **(B)**

FIGURE 10 Artery-like test case: Cross section in the *y*-*z*-plane after (A) extrusion and (B) application of the 4DEMUM.



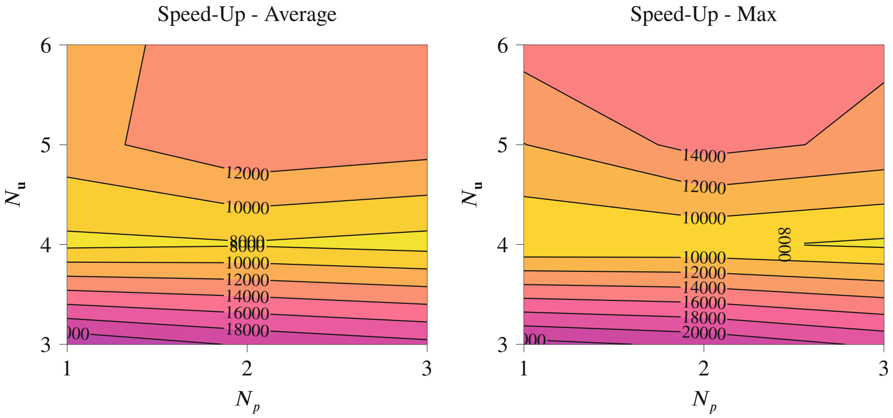
(A) **(B)**

FIGURE 11 Artery-like test case: Results from the construction of the ROM. (A) Distribution of the eigenvalues from the POD. (B) Maximum interpolation error during the EIM.



(A) **(B)**

FIGURE 12 Artery-like test case: Maximum relative error of the ROM over all testing samples. (A) Velocity. (B) Pressure.



(A) **(B)**

FIGURE 13 Artery-like test case: Performance results. (A) Average speed-up over all testing samples. (B) Maximum speed-up over all testing samples.

TABLE 4 Artery-like test case: Values of the runtime of the FOM over the testing set.

	Minimum	Median	Maximum
$T_{\mathrm{CPU}}^{\mathrm{FOM}}$ [s]	98,915.08	100,145.87	104,788.65

Next, we present the results from the performance analysis. The ROM has been evaluated on a single core, whereas we have used 240 cores to compute the FOM. The measured speed-up—the average and the maximum over all the testing samples—is shown in Figure 13. The results are, to some extent, counter-intuitive. First, one can observe that the speed-up is mainly governed by $N_{\boldsymbol{u}}$ and roughly constant for N_p. Second, the speed-up does not change monotonically with increasing $N_{\boldsymbol{u}}$ but has a local minimum at $N_{\boldsymbol{u}} = 4$. Analyzing the evaluation of the ROM in more detail has revealed that this pattern originates from the number of iterations needed to solve the non-linear reduced system. In particular, the value for $N_{\boldsymbol{u}}$ influences the number of non-linear iterations and, in some cases, slows down the solution process. Nevertheless, the order of magnitude of the observed speed-up, ranging from 12,000 to 20,000, is still compelling and even much more significant than in the previous test case. As before, we give information about the absolute runtime of the FOM $T_{\mathrm{CPU}}^{\mathrm{FOM}}$ in Table 4.

So, also the results for this test case underline the advantages of the proposed approach in general and, specifically, its ability to handle problems defined in a four-dimensional space-time domain.

6 | CONCLUSION

In this work, we presented a MOR concept for parametric time-dependent problems defined in domains with prescribed deformations that are allowed to even entail spatial topology changes. There are two main building blocks in this approach. First, we make use of the time-continuous space-time setting. Here, the C-SST-FEM leads to fixed finite-dimensional subspaces for the entire space-time domain, implicitly accounting for current domain deformations. Working on these fixed subspaces, we make use of a projection-based MOR technique based on POD. In contrast to other MOR methods applied to deforming domain problems, the particular approach proposed in this paper can be applied in a straightforward manner. Taken together, we argued that it can reduce the computational complexity for the aforementioned class of problems and, at the same time, maintain the desired level of accuracy in the results.

To confirm this claim, we investigated two representative test cases from the fields of engineering and biomedical applications. In particular, we carried out error and performance analyses for both cases, where we compared the ROM relative to the original time-continuous space-time FOM. For a comparison with respect to a different solver of choice, we additionally provided data to allow a direct conversion of the speed-up results. The first test case was composed of a two-dimensional valve-like geometry containing a moving plug. In the resulting three-dimensional space-time domain, we considered the flow of plastics melt, which occurs in many polymer processing techniques. The parametric character of the problem originated from incorporating uncertainty in the viscosity model, that is, fluctuations in the model parameters. The error analysis showed that we are able to reduce the original model effectively while keeping control over the magnitude of the error in the relevant physical fields. Moreover, a significant speed-up in terms of computational time was proven. By means of the second test case, we extended our analysis to four-dimensional space-time domains that arise when considering three-dimensional bodies in space. Specifically, we studied blood flow in an artery-like volume that undergoes compression, yielding the domain deformation. The problem was varied via a parametric prescribed inflow velocity. The results of the error and performance analysis confirmed the findings from the previous test case and demonstrate the applicability of the approach, also when interested in spatially three-dimensional problems.

Overall, the approach presented here extends the collection of existing MOR techniques with a simple but elegant approach for deforming domain problems. Despite the fact that it shows methodologically favorable characteristics and yields plausible results for the presented test cases, practical difficulties may arise. For example, in three-dimensional application cases, the generation of the required four-dimensional meshes can be challenging and limits the applicability as discussed in Section 3.1. Furthermore, the nature of the resulting reduced system may influence and potentially limit the speed-up that can be realized. Especially, the latter observation remains to be elucidated. Nevertheless, the proposed approach is able to make the advantages of MOR accessible for the class of complex problems that include domain deformations, potentially with spatial topology changes.

ACKNOWLEDGMENTS

Max von Danwitz gratefully acknowledges the funding by dtec.bw—Digitalization and Technology Research Center of the Bundeswehr under the project RISK.twin. dtec.bw is funded by the European Union—NextGenerationEU. Francesco Ballarin and Gianluigi Rozza acknowledge the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA). Francesco Ballarin also thanks the project "Reduced order modeling for numerical simulation of partial differential equations" funded by Università Cattolica del Sacro Cuore, and the INDAM-GNCS project "Metodi numerici per lo studio di strutture geometriche parametriche complesse" (CUP_E53C22001930001, PI Dr Maria Strazzullo). Additionally, Gianluigi Rozza would like to express sincere gratitude to the PRIN NA-FROM-PDEs project for providing the necessary funding and resources. Finally, the authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Fabian Key https://orcid.org/0000-0001-6622-4806
Max von Danwitz https://orcid.org/0000-0002-2814-0027
Francesco Ballarin https://orcid.org/0000-0001-6460-3538

REFERENCES

1. Danwitz VM, Voulis I, Hosters N, Behr M. Time-continuous and time-discontinuous space-time finite elements for advection-diffusion problems. *Int J Numer Methods Eng*. 2023;124(14):3117-3144. doi:10.1002/nme.7241
2. Carreau PJ. Rheological equations from molecular network theories. *Trans Soc Rheol*. 1972;16:99-127. doi:10.1122/1.549276
3. Yasuda K, Armstrong RC, Cohen RE. Shear flow properties of concentrated solutions of linear and star branched polystyrenes. *Rheol Acta*. 1981;20:163-178. doi:10.1007/BF01513059
4. Bird RB, Armstrong RC, Hassager O. *Dynamics of Polymeric Liquids, Volume 1: Fluid Mechanics*. John Wiley and Sons Inc; 1987.
5. Elgeti S, Sauerland H. Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison. *Arch Comput Methods Eng*. 2016;23:323-361. doi:10.1007/s11831-015-9143-2
6. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys*. 1988;79:12-49. doi:10.1016/0021-9991(88)90002-2
7. Chang Y, Hou T, Merriman B, Osher S. A level set formulation of Eulerian Interface capturing methods for incompressible fluid flows. *J Comput Phys*. 1996;124:449-464. doi:10.1006/jcph.1996.0072
8. Hirt C, Nichols B. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys*. 1981;39:201-225. doi:10.1016/0021-9991(81)90145-5
9. Tezduyar T, Behr M, Mittal S, Johnson A. Computation of unsteady incompressible flows with the stabilized finite element methods: space-time formulations, iterative strategies and massively parallel implementations. In: Smolinski P, Liu W, Hulbert G, Tamma K, eds. *New Methods in Transient Analysis*. American Society of Mechanical Engineers; 1992.
10. Batina J. Unsteady Euler Airfoil solutions using unstructured dynamic meshes. *Aiaa J*. 1990;28:1381-1388. doi:10.2514/3.25229
11. Boer dA, Schoot v dM, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct*. 2007;85:784-795. doi:10.1016/j.compstruc.2007.01.013

12. Alauzet F. Efficient moving mesh technique using generalized swapping. In: Jiao X, Weill JC, eds. *Proceedings of the 21st International Meshing Roundtable*. Springer; 2013:17-37.
13. Wang L, Persson PO. A high-order discontinuous Galerkin method with unstructured space-time meshes for two-dimensional compressible flows on domains with large deformations. *Comput Fluids*. 2015;118:53-68. doi:10.1016/j.compfluid.2015.05.026
14. Helmig J, Behr M, Elgeti S. Boundary-conforming finite element methods for twin-screw extruders: unsteady–temperature-dependent–non-Newtonian simulations. *Comput Fluids*. 2019;190:322-336. doi:10.1016/j.compfluid.2019.06.028
15. Hinz J, Helmig J, Möller M, Elgeti S. Boundary-conforming finite element methods for twin-screw extruders using spline-based parameterization techniques. *Comput Methods Appl Mech Eng*. 2020;361:112740. doi:10.1016/j.cma.2019.112740
16. Behr M, Tezduyar T. The shear-slip mesh update method. *Comput Methods Appl Mech Eng*. 1999;174:261-274. doi:10.1016/S0045-7825(98)00299-0
17. Behr M, Tezduyar T. Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components. *Comput Methods Appl Mech Eng*. 2001;190:3189-3200. doi:10.1016/S0045-7825(00)00388-1
18. Behr M, Arora D. Shear-slip mesh update method: implementation and applications. *Comput Methods Biomech Biomed Eng*. 2003;6:113-123. doi:10.1080/1025584031000091650
19. Key F, Pauli L, Elgeti S. The virtual ring shear-slip mesh update method. *Comput Fluids*. 2018;172:352-361. doi:10.1016/j.compfluid.2018.04.006
20. González FA, Elgeti S, Behr M. The surface-reconstruction virtual-region mesh update method for problems with topology changes. *Int J Numer Methods Eng*. 2023;124(9):2050-2067. doi:10.1002/nme.7200
21. Steger J, Benek J. On the use of composite grid schemes in computational aerodynamics. *Comput Methods Appl Mech Eng*. 1987;64:301-320. doi:10.1016/0045-7825(87)90045-4
22. Bazilevs Y, Hughes TJ. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Comput Mech*. 2008;43:143-150. doi:10.1007/s00466-008-0277-z
23. Takizawa K, Tezduyar T, Mochizuki H, et al. Space-time VMS method for flow computations with slip interfaces (ST-SI). *Math Models Methods Appl Sci*. 2015;25:2377-2406. doi:10.1142/S0218202515400126
24. Helmig J, Key F, Behr M, Elgeti S. Combining boundary-conforming finite element meshes on moving domains using a sliding mesh approach. *Int J Numer Methods Fluids*. 2020;93(4):1053-1073. doi:10.1002/fld.4919
25. Peskin C. Flow patterns around heart valves: a numerical method. *J Comput Phys*. 1972;10:252-271. doi:10.1016/0021-9991(72)90065-4
26. Chung J, Hulbert GM. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-alpha method. *J Appl Mech*. 1993;60:371-375. doi:10.1115/1.2900803
27. Hughes TJR, Liu WK, Zimmermann TK. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput Methods Appl Mech Eng*. 1981;29:329-349. doi:10.1016/0045-7825(81)90049-9
28. Donea J, Huerta A, Ponthot JP, Rodríguez-Ferran A. Arbitrary Lagrangian–Eulerian methods. In: Stein E, Borst R, Hughes TJ, eds. *Encyclopedia of Computational Mechanics*. Wiley; 2004.
29. Tezduyar TE, Behr M, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary numerical tests. *Comput Methods Appl Mech Eng*. 1992;94:339-351. doi:10.1016/0045-7825(92)90059-S
30. Tezduyar TE, Behr M, Mittal S, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Comput Methods Appl Mech Eng*. 1992;94:353-371. doi:10.1016/0045-7825(92)90060-W
31. Rendall TCS, Allen CB, Power EDC. Conservative unsteady aerodynamic simulation of arbitrary boundary motion using structured and unstructured meshes in time. *Int J Numer Methods Fluids*. 2012;70:1518-1542. doi:10.1002/fld.2756
32. Danwitz VM, Karyofylli V, Hosters N, Behr M. Simplex space-time meshes in compressible flow simulations. *Int J Numer Methods Fluids*. 2019;91:29-48. doi:10.1002/fld.4743
33. Behr M. Simplex space-time meshes in finite element simulations. *Int J Numer Methods Fluids*. 2008;57:1421-1434. doi:10.1002/fld.1796
34. Karabelas E, Neumüller M. Generating admissible space-time meshes for moving domains in (d + 1) dimensions. In: Langer U, Steinbach O, eds. *Space-Time Methods Berlin*. De Gruyter; 2019.
35. Neumüller M, Steinbach O. Refinement of flexible space–time finite element meshes and discontinuous Galerkin methods. *Comput Vis Sci*. 2011;14:189-205. doi:10.1007/s00791-012-0174-z
36. Caplan PC, Haimes R, Darmofal DL, Galbraith MC. Four-dimensional anisotropic mesh adaptation. *Comput Aided Des*. 2020;129:102915. doi:10.1016/j.cad.2020.102915
37. Danwitz VM, Antony P, Key F, Hosters N, Behr M. Four-dimensional elastically deformed simplex space-time meshes for domains with time-variant topology. *Int J Numer Methods Fluids*. 2021;93:3490-3506. doi:10.1002/fld.5042
38. Hughes TJ, Franca LP. A new finite element formulation for computational fluid dynamics: VII. The stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces. *Comput Methods Appl Mech Eng*. 1987;65:85-96. doi:10.1016/0045-7825(87)90184-8
39. Hughes TJ, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations. *Comput Methods Appl Mech Eng*. 1989;73:173-189. doi:10.1016/0045-7825(89)90111-4
40. Mittal S, Tezduyar TE. Notes on the stabilized space-time finite-element formulation of unsteady incompressible flows. *Comput Phys Commun*. 1992;73:93-112. doi:10.1016/0010-4655(92)90031-S

41. Shakib F, Hughes TJ. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Comput Methods Appl Mech Eng*. 1991;87:35-58. doi:10.1016/0045-7825(91)90145-V
42. Behr M, Tezduyar TE. Finite element solution strategies for large-scale flow simulations. *Comput Methods Appl Mech Eng*. 1994;112:3-24. doi:10.1016/0045-7825(94)90016-7
43. Franca LP, Frey SL. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Comput Methods Appl Mech Eng*. 1992;99:209-233. doi:10.1016/0045-7825(92)90041-H
44. Pauli L. *Stabilized Finite Element Methods for Computational Design of Blood-Handling Devices*. PhD thesis. RWTH Aachen University; 2016.
45. Manzoni A, Quarteroni A, Rozza G. Computational reduction for parametrized PDEs: strategies and applications. *Milan J Math*. 2012;80:283-309. doi:10.1007/s00032-012-0182-y
46. Hesthaven JS, Rozza G, Stamm B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer; 2015.
47. Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng*. 2011;18:395-404. doi:10.1007/s11831-011-9064-7
48. Prud'homme C, Rovas DV, Veroy K, et al. Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods. *J Fluids Eng*. 2002;124:70-80. doi:10.1115/1.1448332
49. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch Comput Methods Eng*. 2008;15:229-275. doi:10.1007/s11831-008-9019-9
50. Rozza G, Huynh D, Nguyen NC, Patera AT. Real-time reliable simulation of heat transfer phenomena. Paper presented at: ASME, ed. Heat Transfer Summer Conference. 2009.
51. Quarteroni A, Rozza G, Manzoni A. Certified reduced basis approximation for parametrized partial differential equations and applications. *J Math Industry*. 2011;1:3. doi:10.1186/2190-5983-1-3
52. Rozza G. Fundamentals of reduced basis method for problems governed by parametrized PDEs and applications. In: Chinesta F, Ladevèze P, eds. *Separated Representations and PGD-Based Model Reduction: Fundamentals and Applications Vienna*. Springer; 2014:153-227.
53. Key F, Ballarin F, Eusterholz S, Elgeti S, Rozza G. Reduced flow model for plastics melt inside an extrusion die. *Proc Appl Math Mech*. 2021;21:e202100071. doi:10.1002/pamm.202100071
54. Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations: an Introduction*. Springer; 2016.
55. Glas S, Mayerhofer A, Urban K. Two ways to treat time in reduced basis methods. In: Benner P, Ohlberger M, Patera A, Rozza G, Urban K, eds. *Model Reduction of Parametrized Systems*. Springer International Publishing; 2017.
56. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *Esaim Math Model Numer Anal*. 2005;39:157-181. doi:10.1051/m2an:2005006
57. Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *Esaim Math Model Numer Anal*. 2008;42:277-302. doi:10.1051/m2an:2008001
58. Haasdonk B. Convergence rates of the pod—greedy method. *Esaim Math Model Numer Anal*. 2013;47:859-873. doi:10.1051/m2an/2012045
59. Drohmann M, Haasdonk B, Ohlberger M. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM*. *J Sci Comput*. 2012;34(2):A937-A969. doi:10.1137/10081157X
60. Sleeman MK, Yano M. Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations. *Comput Methods Appl Mech Eng*. 2022;388:114206. doi:10.1016/j.cma.2021.114206
61. Fick L, Maday Y, Patera AT, Taddei T. A stabilized POD model for turbulent flows over a range of Reynolds numbers: optimal parameter sampling and constrained projection. *J Comput Phys*. 2018;371:214-243. doi:10.1016/j.jcp.2018.05.027
62. Urban K, Patera AT. A new error bound for reduced basis approximation of parabolic partial differential equations. *C R Math*. 2012;350:203-207. doi:10.1016/j.crma.2012.01.026
63. Urban K, Patera AT. An improved error bound for reduced basis approximation of linear parabolic problems. *Math Comput*. 2013;83:1599-1615. doi:10.1090/s0025-5718-2013-02782-2
64. Yano M, Patera AT, Urban K. A space-time hp-interpolation-based certified reduced basis method for burgers' equation. *Math Models Methods Appl Sci*. 2014;24:1903-1935. doi:10.1142/S0218202514500110
65. Yano M. A space-time Petrov Galerkin certified reduced basis method: application to the Boussinesq equations. *SIAM J Sci Comput*. 2014;36(1):A232-A266. doi:10.1137/120903300
66. Choi Y, Carlberg K. Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction. *SIAM J Sci Comput*. 2019;41(1):A26-A58. doi:10.1137/17M1120531
67. Deokar R, Tamma KK. A novel model order reduction framework via staggered reduced basis space-time finite elements in linear first order transient systems. *Int J Heat Mass Transf*. 2018;117:991-1005. doi:10.1016/j.ijheatmasstransfer.2017.10.039
68. Strazzullo M, Ballarin F, Rozza G. POD–Galerkin model order reduction for parametrized time dependent linear quadratic optimal control problems in saddle point formulation. *J Sci Comput*. 2020;83(3):55. doi:10.1007/s10915-020-01232-x
69. Strazzullo M, Ballarin F, Rozza G. POD-Galerkin model order reduction for parametrized nonlinear time-dependent optimal flow control: an application to shallow water equations. *J Numer Math*. 2022;30:63-84. doi:10.1515/jnma-2020-0098
70. Fritzen F, Hassani M. Space–time model order reduction for nonlinear viscoelastic systems subjected to long-term loading. *Meccanica*. 2018;53:1333-1355. doi:10.1007/s11012-017-0734-x
71. Hesthaven JS, Pagliantini C, Rozza G. Reduced basis methods for time-dependent problems. *Acta Numer*. 2022;31:265-345. doi:10.1017/S0962492922000058

72. Urban K. The reduced basis method in space and time: challenges, limits and perspectives. In: Falcone M, Rozza G, eds. *Model Order Reduction and Applications*. Springer Nature Switzerland; 2023:1-72.
73. Anttonen JS. *Techniques for Reduced Order Modeling of Aeoelastic Structures with Deforming Grids*. PhD thesis. Air Force Institute of Technology, Wright-Patterson AFB; 2001.
74. Forti D, Rozza G. Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems. *Int J Comut Fluid Dyn*. 2014;28:158-169. doi:10.1080/10618562.2014.932352
75. Izadi M, Dubljevic S. Order-reduction of parabolic PDEs with time-varying domain using empirical eigenfunctions. *AIChE J*. 2013;59:4142-4150. doi:10.1002/aic.14152
76. Narasingam A, Siddhamshetty P, Kwon JS. Temporal clustering for order reduction of nonlinear parabolic PDE systems with time-dependent spatial domains: application to a hydraulic fracturing process. *AIChE J*. 2017;63:3818-3831. doi:10.1002/aic.15733
77. Ballarin F, Rebollo TC, Ávila ED, Mármol MG, Rozza G. Certified reduced basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height. *Comput Math Appl*. 2020;80:973-989. doi:10.1016/j.camwa.2020.05.013
78. Drohmann M, Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximation of evolution equations on parametrized geometries. In: Handlovičová A, Frolkovič P, Mikula K, Ševčovič D, eds. *Proceedings of ALGORITMY*. Slovak University of Technology in Bratislava; 2009.
79. Fröhlich B, Gade J, Geiger F, Bischoff M, Eberhard P. Geometric element parameterization and parametric model order reduction in finite element based shape optimization. *Comput Mech*. 2019;63:853-868. doi:10.1007/s00466-018-1626-1
80. Karatzas EN, Stabile G, Nouveau L, Scovazzi G, Rozza G. A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a stokes flow. *Comput Methods Appl Mech Eng*. 2019;347:568-587. doi:10.1016/j.cma.2018.12.040
81. Lassila T, Rozza G. Parametric free-form shape design with PDE models and reduced basis method. *Comput Methods Appl Mech Eng*. 2010;199:1583-1592. doi:10.1016/j.cma.2010.01.007
82. Manzoni A, Quarteroni A, Rozza G. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *Int J Numer Methods Fluids*. 2012;70:646-670. doi:10.1002/fld.2712
83. Mola A, Demo N, Tezzele M, Rozza G. Geometrical parametrization and morphing techniques with applications. In: Rozza G, Stabile G, Ballarin F, eds. *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. Society for Industrial and Applied Mathematics; 2022.
84. Salmoiraghi F, Scardigli A, Telib H, Rozza G. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *Int J Comput Fluid Dyn*. 2018;32:233-247. doi:10.1080/10618562.2018.1514115
85. Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C R Math*. 2004;339:667-672. doi:10.1016/j.crma.2004.08.006
86. Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Esaim Math Model Numer Anal*. 2007;41:575-605. doi:10.1051/m2an:2007031
87. Sirovich L. Turbulence and the dynamics of coherent structures. *I Coherent Structures Q Appl Math*. 1987;45:561-571. doi:10.1090/qam/910462
88. Schröder T. *Rheologie der Kunststoffe: Theorie Und Praxis*. Carl Hanser Verlag GmbH Co KG; 2020.
89. Cho YI, Kensey KR. Effects of the non-Newtonian viscosity of blood on flows in a diseased arterial vessel. Part 1: steady flows. *Biorheology*. 1991;28:241-262. doi:10.3233/BIR-1991-283-415

How to cite this article: Key F, von Danwitz M, Ballarin F, Rozza G. Model order reduction for deforming domain problems in a time-continuous space-time setting. *Int J Numer Methods Eng*. 2023;124(23):5125-5150. doi: 10.1002/nme.7342

APPENDIX A. ANALYSIS OF THE INFLUENCE OF THE AMOUNT OF TRAINING DATA

In the appendix of this work, we present a brief analysis of the influence of the amount of training data on the accuracy of the ROM. For the test cases presented in Section 5, we perform numerical experiments in which we reduce the number of training samples from the quantity initially used. By investigating the behavior of the ROMs across different training data sizes, we aim to gain deeper insights regarding the expected offline costs and the generalization of the proposed methodology.

A.1 Valve-like test case

For the valve-like test case, we choose $N_{\text{train}} \in \{4, 16, 64, 256\}$ where $N_{\text{train}} = 256$ was used in Section 5.1.1. As before, the training samples are equidistantly spaced for each N_{train}. The results for the normalized eigenvalues and the EIM error are given in Figure A1. Both show a consistent behavior for all numbers of training samples, with the only distinction

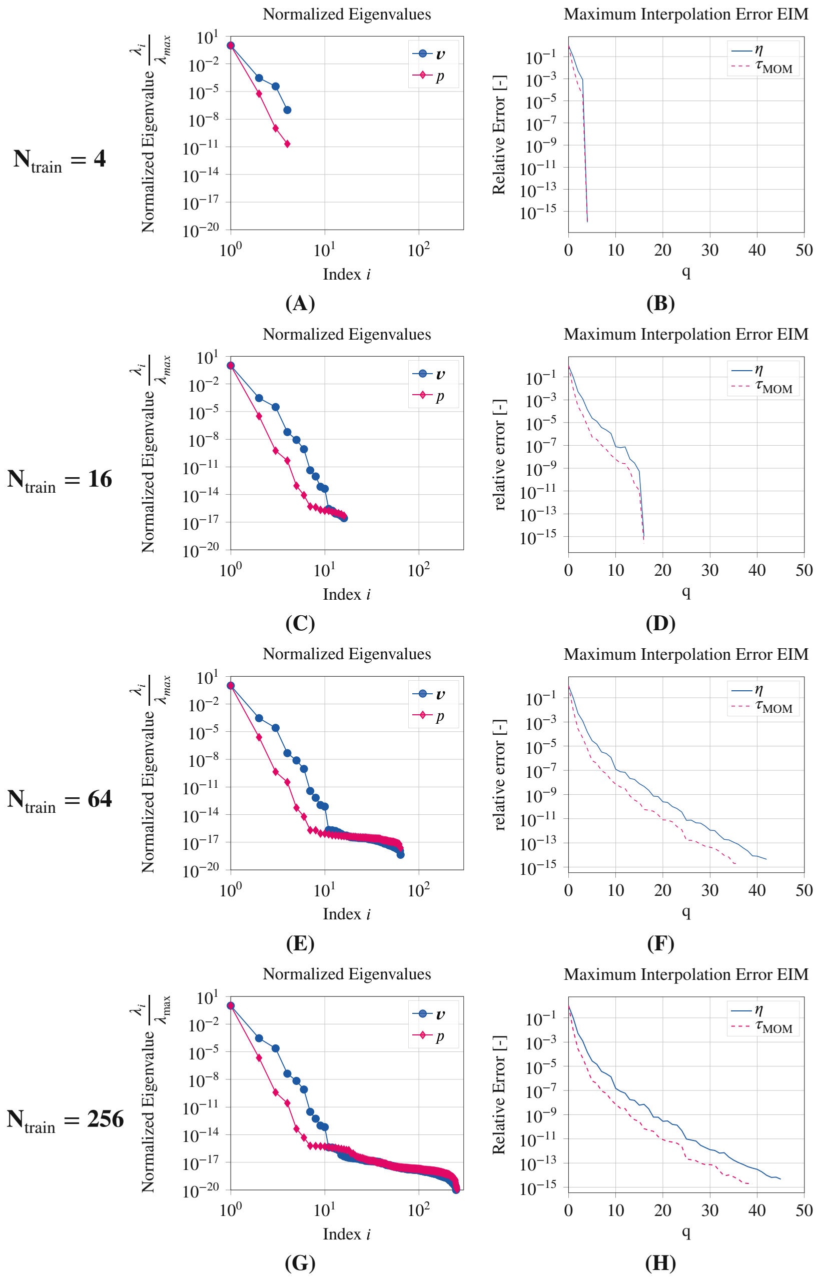


FIGURE A1 Valve-like test case: Results from the construction of the ROM for different numbers of training samples N_{train} (rows); distribution of the eigenvalues from the POD (left column) and maximum interpolation error during the EIM (right column).

being that each curve is truncated after reaching its respective maximum number of samples. For the EIM, note that the error immediately drops to machine accuracy as soon as $q = N_{\text{train}}$, since it is able to interpolate all of the training points. Given the eigenvalues, we choose the maximal number of basis functions for the velocity and pressure, that is, $N_{\boldsymbol{u}}^{\max}$ and $N_p^{\max}$, based on the retained energy (see Table A1).

In Figure A2, we show the velocity and pressure errors for different numbers of training samples. In these plots, white areas appear when the maximum number of basis functions for the ROM has been exceeded. One can observe that the errors are qualitatively the same for each choice of N_{train}, suggesting that a smaller number of training samples is sufficient to obtain a ROM with comparable accuracy. However, it can also be seen that the choice of $N_{\text{train}} = 4$ does not allow to reduce the errors as much as in the other cases. Furthermore, the eigenvalue distribution seems to be an adequate indication of the minimum number of training samples. Nevertheless, we would like to reiterate that this is the result of a post-analysis and cannot be determined in advance.

TABLE A1 Valve-like test case: Maximal numbers of basis functions for different sizes of the training set.

N_{train}	4	16	64	256
$N_{\boldsymbol{u}}^{\max}$	5	11	11	11
$N_p^{\max}$	4	6	5	6

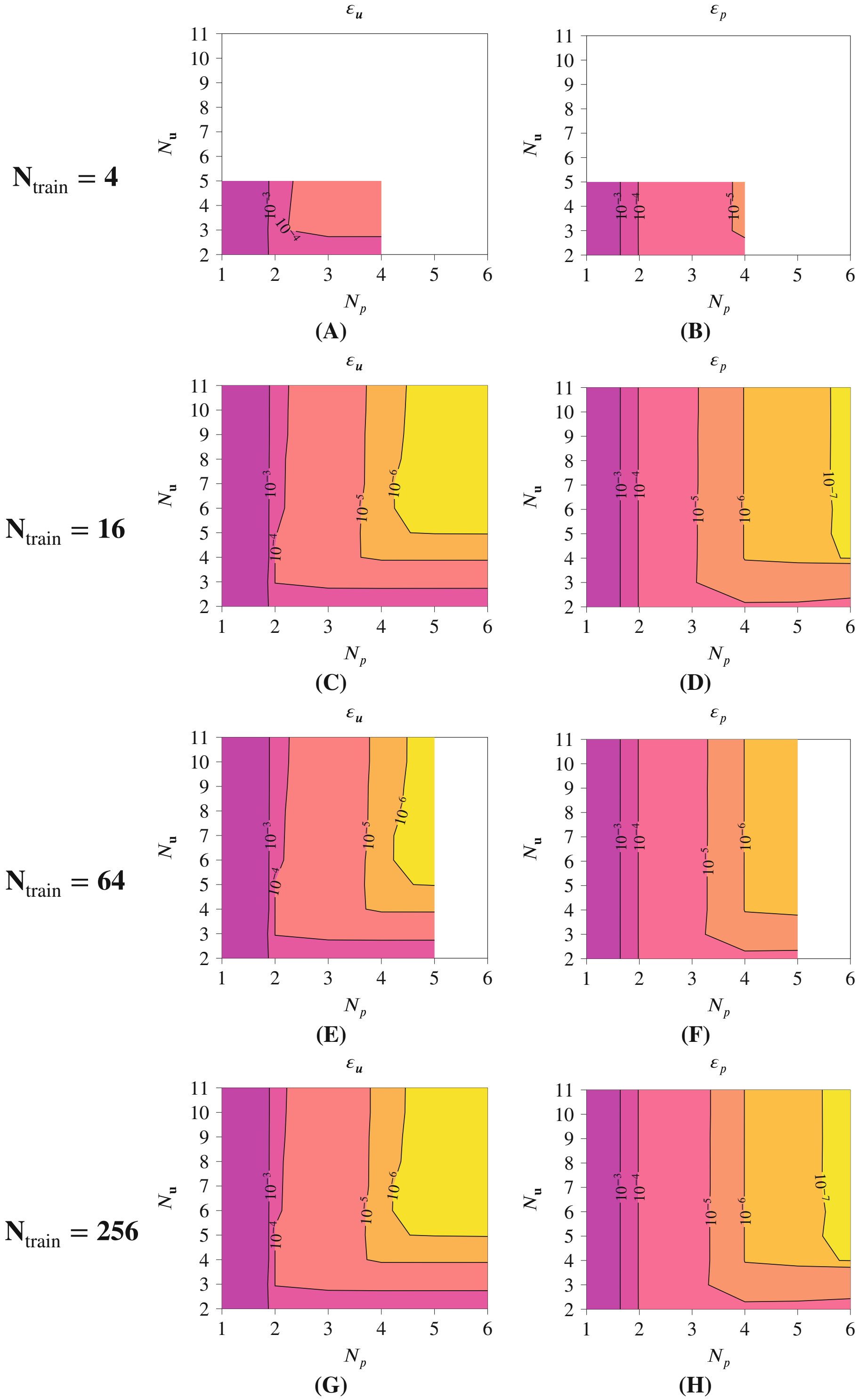
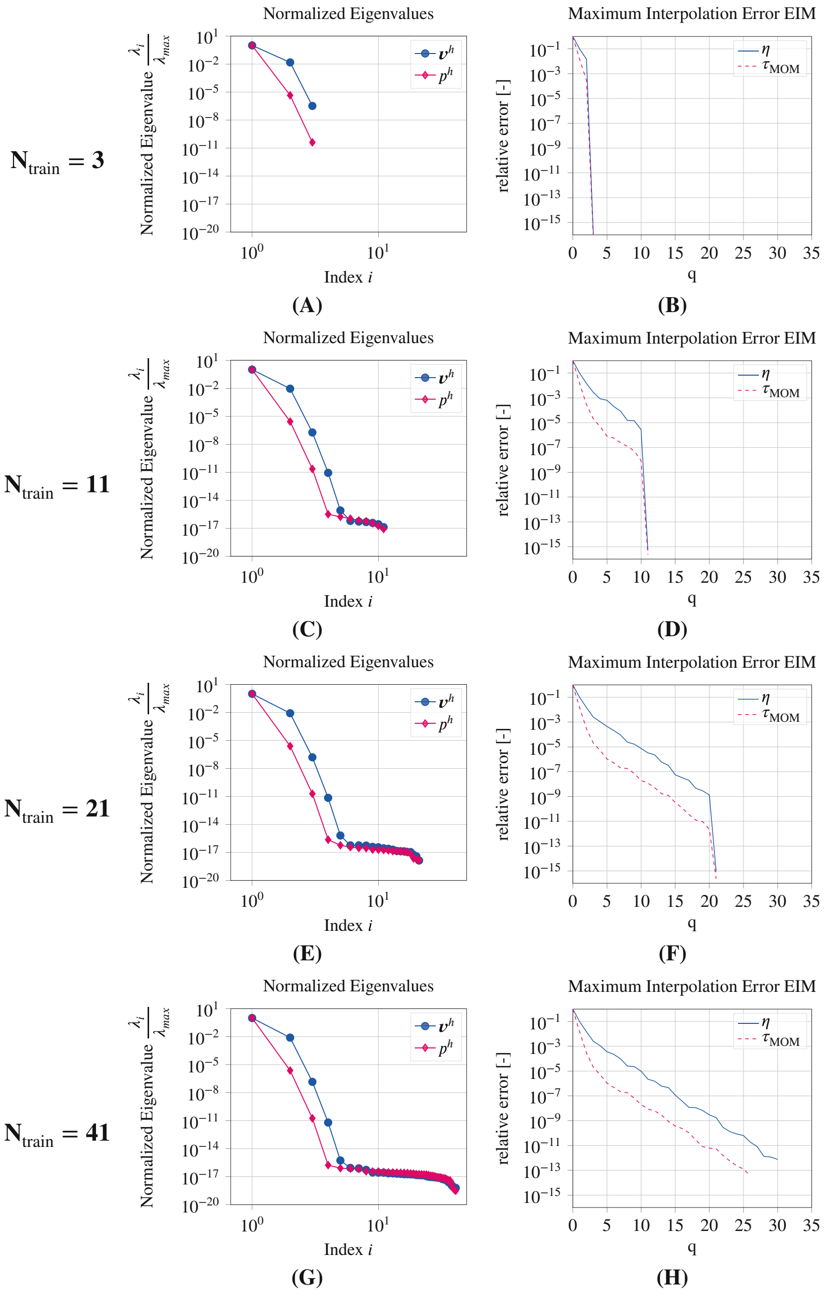


FIGURE A2 Valve-like test case: Maximum relative error of the ROM over all testing samples for different numbers of training samples N_{train} (rows) and for velocity (left column) and pressure (right column).

A.2 Artery-like test case

The same experiment has been performed for the artery-like test case. Here, we use $N_{\text{train}} \in \{3, 11, 21, 41\}$ equidistant training samples, where $N_{\text{train}} = 41$ is the original number chosen in Section 5.2.1. In Figure A3, the eigenvalues and EIM errors are depicted. It can be seen that the curves show the same characteristics as previously observed in the valve-like test case. Regardless of the number of training samples, the behavior remains qualitatively consistent, exhibiting the distinctive cut-off. The maximal numbers of basis functions that were used to construct the respective ROMs are given in Table A2.

Figure A4 presents the relative error in the velocity and pressure fields for the different values of N_{train}. Again, we observe that the error exhibits a similar behavior even with a decreased amount of training data. It is only when we reduce N_{train} too much, that is, for $N_{\text{train}} = 3$, that we lose an order of magnitude in the minimum errors. Also for this test case, the behavior corresponds to the distribution of eigenvalues.



(A) (B)

(C) (D)

(E) (F)

(G) (H)

FIGURE A3 Artery-like test case: Results from the construction of the ROM for different numbers of training samples N_{train} (rows); distribution of the eigenvalues from the POD (left column) and maximum interpolation error during the EIM (right column).

TABLE A2 Artery-like test case: Maximal numbers of basis functions for different sizes of the training set.

N_{train}	3	11	21	41
$N_{\boldsymbol{u}}^{\max}$	5	6	6	6
$N_p^{\max}$	3	3	3	3

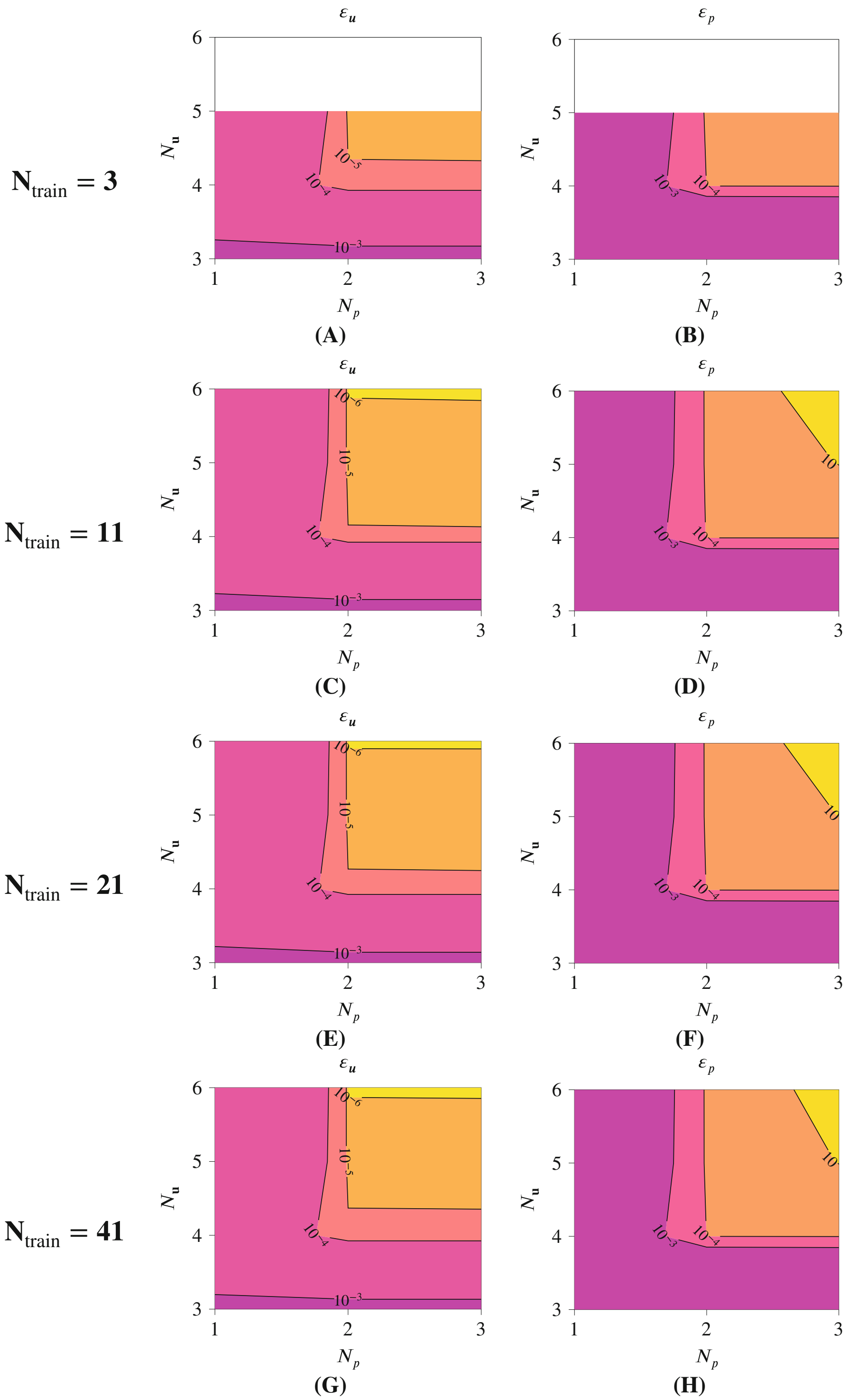


FIGURE A4 Artery-like test case: Maximum relative error of the ROM over all testing samples for different numbers of training samples N_{train} (rows) and for velocity (left column) and pressure (right column).