Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 346 (2019) 486–512
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Fast divergence-conforming reduced basis methods for steady Navier–Stokes flow

Eivind Fonn[a,*], Harald van Brummelen[b], Trond Kvamsdal[a,c], Adil Rasheed[a]

[a] *SINTEF Digital, Trondheim, Norway*
[b] *Department of Mechanical Engineering, Eindhoven University of Technology, Netherlands*
[c] *Department of Mathematical Sciences, Norwegian University of Science and Technology, Norway*

Received 16 July 2018; received in revised form 26 November 2018; accepted 29 November 2018
Available online 17 December 2018

Highlights

- Isogeometric methods enable velocity-only divergence-free reduced models.
- Such divergence-free reduced models are considerably faster in the only stage.
- Parametrized geometry variation can be handled with the Piola transform.
- Pressure fields can be reconstructed with negligible computational expense.

Abstract

Reduced-basis methods (RB methods or RBMs) form one of the most promising techniques to deliver numerical solutions of parametrized PDEs in real-time with reasonable accuracy. For incompressible flow problems, RBMs based on LBB stable velocity–pressure spaces do not generally inherit the stability of the underlying high-fidelity model and, instead, additional stabilization techniques must be introduced. One way of bypassing the loss of LBB stability in the RBM is to inflate the velocity space with supremizer modes. This however deteriorates the performance of the RBM in the performance-critical online stage, as additional DOFs must be introduced to retain stability, while these DOFs do not effectively contribute to accuracy of the RB approximation. In this work we consider a velocity-only RB approximation, exploiting a solenoidal velocity basis. The solenoidal reduced basis emerges directly from the high-fidelity velocity solutions in the offline stage. By means of Piola transforms, the solenoidality of the velocity space is retained under geometric transformations, making the proposed RB method suitable also for the investigation of geometric parameters. To ensure exact solenoidality of the high-fidelity velocity solutions that constitute the RB, we consider approximations based on divergence-conforming compatible B-splines. We show that the velocity-only RB method leads to a significant improvement in computational efficiency in the online stage, and that the pressure solution can be recovered a posteriori at negligible extra cost. We illustrate the solenoidal RB approach by modeling steady two-dimensional Navier–Stokes flow around a NACA0015 airfoil at various angles of attack.

Keywords: Reduced order modeling; Reduced basis method; Isogeometric analysis; Divergence-conforming

* Corresponding author.
E-mail address: eivind.fonn@sintef.no (E. Fonn).

https://doi.org/10.1016/j.cma.2018.11.038

1. Introduction

Conventional methods for simulating partial differential equations include well-established techniques such as Finite Volume Methods (FVM), Finite Difference Methods (FDM) and Finite Element Methods (FEM). Common to all of these methods is the large number of degrees of freedom that is typically required to accurately model a physical system, often numbering in the millions or billions. Given the good and well-established approximation properties of FVM, FDM and in particular FEM, such models are usually classified as *high-fidelity* models. Problems of this size are generally not possible to solve in realistic timeframes except on specialized high-performance computing facilities, which renders such computations very expensive, and even then they may require several days of computing time. In addition, the need for dedicated high-end computing resources prohibits the use of high-fidelity models in time-critical on-site analyses such as biomedical applications, control strategies and hybrid analytics, in view of the need for additional special infrastructure and the incurrence of further communication overheads.

The excessive computational complexity of high-fidelity models is also at odds with the increasing demand for real-time low-cost models for the repetitive solution of partial differential equations (PDEs) in many-query scenarios. This is particularly relevant in optimization, control systems, inverse and inference problems, and uncertainty quantification. Common for many of these applications is that the PDE in question is *parametrized* by a suitably small number of input parameters. It is often necessary to demand solve times in the sub-second regime, which is entirely infeasible with conventional methods.

Reduced-order modeling (ROM) provides a paradigm to address the aforementioned challenges. ROM is a rapidly developing field [1]. The general aim of ROM is to replace the original model with a *reduced model* of very modest computational complexity. Within this general framework of ideas, we can distinguish between *a priori* type methods, e.g. the Proper Generalized Decomposition (PGD) method, see for example [2–5], and *a posteriori* type methods, based on the Proper Orthogonal Decomposition (POD). In this paper we shall focus on the latter, specifically *reduced basis methods* (RBM). These methods date back to the 1980s with work from [6–12] and the first theoretical foundations of the method were given by Fink and Rheinboldt in [13,14]. It was first used for flow problems in work by Peterson and Gunzburger [15,16], and much of the more recent work can be attributed to the theoretical foundations in [17,18]. Excellent modern introductions can be found in [19,20].

The fundamental concept of reduced basis methods is to formulate the problem on a function space with very low dimension that is tailored to the solution of the PDE in the parameter regime of interest. In comparison, while finite-element spaces generally have well-established asymptotic approximation properties, e.g. density and quasi-optimality for smooth functions, and the computational cost of constructing a FEM basis function is generally negligible, the approximation properties of a FEM basis per-degree-of-freedom (i.e. for each of the basis functions separately) are clearly limited. RB methods seek to construct an approximation space that has optimal approximation properties with respect to dimension for a restricted class of functions, viz. the solutions of the PDE in the parameter space under consideration, under the premise that the cost of constructing the basis is inconsequential. The latter premise arises because the construction of the basis is carried out a priori in the so-called offline stage. To do this, some high-fidelity solutions must be computed in advance, called an *ensemble* of *snapshots*.

RB methods are strictly divided in two stages, viz. the *offline* and *online* stages. The offline stage is run only once, and the online stage is run once for each problem instance to solve (that is to say, for each parameter query). The goal is then to off-load as much work as possible from the online to the offline stage, so that the total cost per online execution is very small in an amortized sense. In other words, work in the offline stage is considered "free" and all efforts will be focused on creating a cheap online stage.

RB methods have been applied in various application areas, e.g. optimal control [21,22], inverse problems [23,24], shape optimization [25,26], quantum models [27] and solid mechanics [9,28]. Development and application of RB methods to flow problems have been considered in for instance [15,16,29,30]. The principal novelty of the present paper is the development of a solenoidal (divergence-free) velocity-only RB method for incompressible-fluid flow problems with geometric parametrizations, based on divergence-conforming high-fidelity approximations. The advantage of the solenoidal-velocity RBM is that the pressure variable is redundant and can be ignored in the online stage. The absence of the pressure variable in the RBM in fact carries a two-fold benefit. First, velocity–pressure RBs do not generally inherit the inf–sup stability of the underlying high-fidelity models. Instability of the RB velocity–pressure pair can be resolved by extending the velocity space with additional supremizer modes [30,31]. This however introduces additional DOFs in the performance-critical online stage, while these DOFs do not effectively contribute to accuracy of the RB approximation. By incorporating the solenoidality constraint in the RB, the approximate

formulation is released of its mixed-character, and inf–sup stability considerations for the RBM are voided. Second, the removal of the pressure variable in the RB reduces the DOF count in the online stage, accelerating the solution process in the time-critical phase.

The use of solenoidal ROM for incompressible-flow problems has been propounded before; see, e.g. [31,32]. However, it appears that the concept was not followed through for the important scenario of geometric parametrizations, in view of the complexity pertaining to the required Piola transforms in the ROM. For physical parameters, solenoidal RB is in fact straightforward. For parameters related to essential boundary conditions, solenoidal RB is more complicated by the fact that solenoidal lifts of the boundary data are required. Geometry parametrizations in addition require that the solenoidality property of the RB is retained under the considered geometric transformations. A central component in a solenoidal RB method is therefore the Piola transform, which provides solenoidality-preserving transformations. The Piola transform however leads to complicated non-polynomial expressions in the ROM, for which it is not generally possible to construct the affine representations that are required for an efficient online stage. In this work we consider truncated series expansions of these expressions that do admit an affine representation, and we assess the computational efficiency of such an approach.

An important additional aspect of the present work is the use of divergence-conforming B-spline approximations for the high-fidelity simulations. Such methods were first introduced by Buffa and co-workers [33,34] and then by Evans and Hughes [35–38]. Extension to locally refined B-splines [39,40] was done in [41] and to divergence-conforming multiscale turbulence models in [42]. They rely on formulating a pressure space that is exactly equal to the divergence of the velocity space. This is enabled by the use of B-spline basis functions, recently popularized in the field of isogeometric analysis [43]. The divergence-conforming approximations provide pointwise solenoidal snapshots, as opposed to the weakly solenoidal high-fidelity solutions provided by, for instance, conventional Taylor–Hood elements. The Piola transform in turn ensures that this property is retained in the solenoidal RB method.

The paper is organized as follows:

- Section 2 formulates the classical stationary Navier–Stokes problem in a Finite Element setting, introduces the notation needed for handling parametrized versions of the problem, and introduces the critically important (to Reduced Basis Methods) affine representations in Section 2.4.
- Section 3 describes the reduced basis method as applied to the Navier–Stokes problem in detail. One method for pressure stabilization is introduced in Section 3.4.
- Section 4 introduces the main novelty of the paper, the application of divergence-conforming high-fidelity methods to a reduced basis method.
- Section 5 develops the necessary mathematics for a numerical example: flow around a NACA0015 airfoil, parametrized by angle-of-attack and inflow velocity. This development is made in parallel for two methods: a divergence-conforming reduced basis method, and a conventional one (read: not divergence-conforming).
- Section 6 presents the most important results from this example. Particular emphasis is made on evaluating the convergence properties of the reduced methods (that is, the degree to which their solutions agree with the corresponding high-fidelity method) and their speed of execution.
- Finally, Section 7 summarizes the findings.

2. Stationary Navier–Stokes problem

2.1. Weak formulation

We consider the stationary Navier–Stokes equations,

$$-\nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \Gamma_{\mathrm{D}}, |\Gamma_D| > 0 \tag{3}$$

$$-p\boldsymbol{n} + \nu(\nabla\boldsymbol{u})\boldsymbol{n} = \boldsymbol{h} \qquad \text{on } \Gamma_{\mathrm{N}}. \tag{4}$$

where ν is the viscosity, $\boldsymbol{u}$, p are the unknown velocity and pressure, $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{h}$ correspond to exogenous data, $\Omega \subset \mathbb{R}^d$ is the domain of interest with boundary $\partial\Omega = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}}$, $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$ and $\boldsymbol{n}$ denotes the external unit normal vector.

We define the function spaces

$$U_\zeta = \left\{ \boldsymbol{u} \in [H^1(\Omega)]^d \mid \boldsymbol{u} = \zeta \text{ on } \Gamma_{\mathrm{D}} \right\}, \tag{5}$$

$$P = L^2(\Omega). \tag{6}$$

and note that the weak Galerkin formulation of the problem is to find $(\boldsymbol{u}, p) \in (U_{\boldsymbol{g}}, P)$ such that, for all $(\boldsymbol{w}, q) \in (U_0, P)$ it holds that

$$a(\boldsymbol{u}, \boldsymbol{w}) + c(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{w}) + b(p, \boldsymbol{w}) = d(\boldsymbol{w}), \tag{7}$$

$$b(q, \boldsymbol{u}) = 0, \tag{8}$$

where the linear, bilinear and trilinear forms a, b, c, d are defined as

$$a(\boldsymbol{u}, \boldsymbol{w}) = \nu \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{w}, \tag{9}$$

$$b(p, \boldsymbol{w}) = -\int_\Omega p \nabla \cdot \boldsymbol{w}, \tag{10}$$

$$c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \int_\Omega (\boldsymbol{u} \cdot \nabla) \boldsymbol{v} \cdot \boldsymbol{w}. \tag{11}$$

$$d(\boldsymbol{w}) = \int_{\Gamma_{\mathrm{N}}} \boldsymbol{h} \cdot \boldsymbol{w} + \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w}, \tag{12}$$

For the purposes of solving (7)–(8), it is customary to introduce a *lift function* $\ell \in U$ satisfying $\boldsymbol{\ell} = \boldsymbol{g}$ on Γ_{D} and to solve for the difference $\boldsymbol{u} - \ell$, which satisfies homogeneous boundary conditions and thus lives in a linear space, as opposed to $\boldsymbol{u}$, which resides in an affine space. To this end, write $\boldsymbol{u} = \boldsymbol{u}_0 + \ell$. The modified problem then reads: find $(\boldsymbol{u}_0, p) \in (U_0, P)$ such that, for all $(\boldsymbol{w}, q) \in (U_0, P)$ it holds that

$$a(\boldsymbol{u}_0, \boldsymbol{w}) + c(\boldsymbol{u}_0, \boldsymbol{u}_0, \boldsymbol{w}) + c_0(\boldsymbol{u}_0, \boldsymbol{w}) + b(p, \boldsymbol{w}) = d_0(\boldsymbol{w}) \tag{13}$$

$$b(q, \boldsymbol{u}_0) = d_1(q). \tag{14}$$

where the new forms read

$$c_0(\boldsymbol{u}, \boldsymbol{w}) = c(\boldsymbol{u}, \boldsymbol{\ell}, \boldsymbol{w}) + c(\boldsymbol{\ell}, \boldsymbol{u}, \boldsymbol{w}) \tag{15}$$

$$d_0(\boldsymbol{w}) = d(\boldsymbol{w}) - c(\boldsymbol{\ell}, \boldsymbol{\ell}, \boldsymbol{w}) - a(\boldsymbol{\ell}, \boldsymbol{w}), \tag{16}$$

$$d_1(q) = -b(q, \boldsymbol{\ell}) \tag{17}$$

One may then recover the original through $\boldsymbol{u} = \boldsymbol{u}_0 + \ell$. The lift-based procedure for treating Dirichlet boundary conditions is well-known, and most FEM implementations handle boundary conditions automatically by categorizing degrees of freedom as internal (free) and boundary (constrained), requiring no cognitive overhead for the user. However, it is worth alluding to this point here since the reduced-basis method presupposes homogeneous Dirichlet boundary conditions, in order to form a linear space.

2.2. Velocity-only formulation

For what follows it is worth considering the effects of choosing divergence-free velocity function spaces. In particular, we will assume in this subsection that $\nabla \cdot \boldsymbol{u}_0 = 0$ for all $\boldsymbol{u}_0 \in U_0$, and also that $\nabla \cdot \ell = 0$. This ensures that $\nabla \cdot \boldsymbol{u} = 0$, so the continuity equation (2) and its variational counterpart (14) are both satisfied by construction. The momentum equation (13) then reduces to

$$a(\boldsymbol{u}_0, \boldsymbol{w}) + c(\boldsymbol{u}_0, \boldsymbol{u}_0, \boldsymbol{w}) + c_0(\boldsymbol{u}_0, \boldsymbol{w}) = d_0(\boldsymbol{w}) \tag{18}$$

In other words, we have a *velocity-only* formulation.

Considering numerical approximations of (1)–(4), formulation (18) has some practical benefits: in view of the absence of the pressure variable, it voids all concerns about pressure stability, and it reduces the size of the discretization. On the other hand, two problems arise. The first problem is that it is practically infeasible to create divergence-free function spaces, which is why this formulation does not see much use in conventional approximation

methods. The second problem is that the pressure is often of interest, e.g. in post-processing operations to determine flow-induced loads by means of extraction functions [44,45], depreciating a solver based on (18) in many applications.

The proposed solenoidal RB method forms an instance of the velocity-only formulation (18). However, the problems mentioned above for conventional approximation methods are mitigated in the RBM setting. First, the inherent solenoidality of the snapshots facilitates the construction of a divergence-free *reduced basis*. Each snapshot corresponds to a high-fidelity solution of (13)–(14) and, hence, the computational cost of constructing the solenoidal basis is indeed significant. However, the computational cost of constructing snapshots is inherent to RBM, and it is ignored because this expense is incurred in the offline stage. Second, the pressure solution can be recovered *a posteriori*, by solving (13) with a different test space, to find $p \in P$ such that for all $\boldsymbol{s} \in S$ it holds that

$$b(p, \boldsymbol{s}) = d_0(\boldsymbol{s}) - a(\boldsymbol{u}_0, \boldsymbol{s}) - c(\boldsymbol{u}_0, \boldsymbol{u}_0, \boldsymbol{s}) - c_0(\boldsymbol{u}_0, \boldsymbol{s}), \tag{19}$$

where $\boldsymbol{u}_0$ is the RB velocity solution. By virtue of the absence of the pressure variable in the momentum equation (18) in the velocity-only formulation, the reduced pressure space can in principle be selected arbitrarily, independent of the velocity space. Noting that the high-fidelity solutions according to (13)–(14) do comprise a pressure component, a plausible reduced pressure space for (19) can in fact be directly extracted from the pressure components of the high-fidelity solutions; see Section 3.3. Subsequent to selection of the reduced pressure space, the test space S should be chosen such that the left-hand-side of (19) becomes suitably non-singular. Section 3.4 elaborates on a procedure to this purpose.

2.3. Parametric dependence

We consider the case where the problem (7)–(8) depends on a number of *parameters*. Denote by $\mathcal{P}$ the parameter space (which will for the moment remain abstract) and by μ any given element of $\mathcal{P}$. The typical parameters of interest are generally grouped into classes: *physical parameters*, *data parameters* and *geometric parameters*. The purpose of the following discussion is to explicitly resolve the parametric dependence of the multi-linear forms. We shall denote this with notation such as $a(\boldsymbol{u}, \boldsymbol{w}; \mu)$, and similarly for other forms.

Loosely stated, physical parameters are those parameters impacting the physical quantities involved in the model, e.g. the viscosity ν in (1). Data parameters impact the source term $\boldsymbol{f}$, and Dirichlet or Neumann boundary data, i.e. the functions $\boldsymbol{g}$ and $\boldsymbol{h}$ in (3)–(4), and through $\boldsymbol{g}$ also the lifting function ℓ. The aforementioned types of parameters appear directly in the multi-linear forms, viz.

$$a(\boldsymbol{u}, \boldsymbol{w}; \mu) = \nu(\mu) \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{w}, \tag{20}$$

$$d_0(\boldsymbol{w}; \mu) = \int_{\Gamma_\mathrm{N}} \boldsymbol{h}(\mu) \cdot \boldsymbol{w} + \int_\Omega \boldsymbol{f}(\boldsymbol{\mu}) \cdot \boldsymbol{w}, \tag{21}$$

$$c_0(\boldsymbol{u}, \boldsymbol{w}; \mu) = c(\boldsymbol{u}, \ell(\mu), \boldsymbol{w}) + c(\ell(\mu), \boldsymbol{u}, \boldsymbol{w}), \tag{22}$$

$$d_1(q; \mu) = -b(q, \ell(\mu)). \tag{23}$$

One may note that while the original variational form (7)–(8) is set in a function space that is possibly parameter-dependent via the Dirichlet data, the corresponding homogeneous variational form (13)–(14) is in canonical form in the sense that the ambient spaces are parameter independent, and the parameter dependence occurs only in the functionals.

Let us next consider geometric parameters, by which we mean that the computational domain $\Omega = \Omega(\mu)$ varies in relation to the parameters. To cast this problem in canonical form, i.e. to transfer the parameter dependence from the ambient spaces to the functionals in the weak formulation, we pull back the corresponding multi-linear forms to (function spaces on) a fixed *reference domain*. Let $\hat{\Omega}$ denote a suitable reference domain, (usually $\hat{\Omega} = \Omega(\boldsymbol{\mu})$ for some μ, but this is not necessarily the case), such that there exists a bi-Lipschitz bijection $\chi_{\boldsymbol{\mu}} : \Omega(\mu) \to \hat{\Omega}$, which maps any of the parametrized domains to the reference domain. We insist that the images of Γ_D and Γ_N are fixed in $\hat{\Omega}$. In other words, it holds that

$$\chi_{\boldsymbol{\mu}}(\Gamma_\mathrm{D}(\mu)) = \hat{\Gamma}_\mathrm{D} \subset \partial \hat{\Omega} \tag{24}$$

and that $\hat{\Gamma}_{\mathrm{D}}$ as defined is independent of μ. Based on the domain map $\chi_{\boldsymbol{\mu}}$, we introduce two homeomorphisms $\pi_\mu^{\mathrm{V}} : [H^1(\hat{\Omega})]^d \to [H^1(\Omega(\mu))]^d$ and $\pi_\mu^{\mathrm{P}} : L^2(\hat{\Omega}) \to L^2(\Omega(\boldsymbol{\mu}))$. For instance, $\boldsymbol{\pi}_\mu := (\pi_\mu^{\mathrm{V}}, \pi_\mu^{\mathrm{P}})$ can be defined as the canonical transformation:

$$\boldsymbol{\pi}_\mu : (\hat{\boldsymbol{u}}, \hat{p}) \mapsto \big(\hat{\boldsymbol{u}} \circ \chi_\mu, \hat{p} \circ \chi_\mu\big) \tag{25}$$

However, alternate maps can be constructed. The pullback of the multi-linear forms a, b, c, c_0 (to the reference domain, by the map $\boldsymbol{\pi}_\mu$) is then defined by:

$$\begin{aligned}
(\pi_\mu^* a)\big(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \boldsymbol{\mu}\big) &= a\big(\pi_\mu^{\mathrm{V}}\hat{\boldsymbol{u}}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{w}}; \boldsymbol{\mu}\big)\\
(\pi_\mu^* b)\big(\hat{p}, \hat{\boldsymbol{w}}; \boldsymbol{\mu}\big) &= b\big(\pi_\mu^{\mathrm{P}}\hat{p}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{w}}; \boldsymbol{\mu}\big)\\
(\pi_\mu^* c)\big(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}; \boldsymbol{\mu}\big) &= c\big(\pi_\mu^{\mathrm{V}}\hat{\boldsymbol{u}}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{v}}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{w}}; \boldsymbol{\mu}\big)\\
(\pi_\mu^* c_0)\big(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \boldsymbol{\mu}\big) &= c\big(\pi_\mu^{\mathrm{V}}\hat{\boldsymbol{u}}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{\ell}}(\boldsymbol{\mu}), \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{w}}; \boldsymbol{\mu}\big) + c\big(\pi_\mu^{\mathrm{V}}\hat{\boldsymbol{\ell}}(\boldsymbol{\mu}), \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{u}}, \pi_\mu^{\mathrm{V}}\hat{\boldsymbol{w}}; \boldsymbol{\mu}\big)
\end{aligned} \tag{26}$$

and the pullback of the linear forms d_0 and d_1 is defined analogously. By means of the pullback operation π_μ^*, the parametric version of (13)–(14) can be cast in the canonical form: find $(\hat{\boldsymbol{u}}_0, \hat{p}) \in (\hat{U}_0, \hat{P})$ such that, for all $(\hat{\boldsymbol{w}}, \hat{q}) \in (\hat{U}_0, \hat{P})$ it holds that

$$(\pi_\mu^* a)(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{w}}; \mu) + (\pi_\mu^* c)(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{u}}_0, \hat{\boldsymbol{w}}; \mu) + (\pi_\mu^* c_0)(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{w}}; \mu) + (\pi_\mu^* b)(\hat{p}, \hat{\boldsymbol{w}}; \mu) = (\pi_\mu^* d_0)(\hat{\boldsymbol{w}}; \mu), \tag{27}$$

$$(\pi_\mu^* b)(\hat{q}, \hat{\boldsymbol{u}}_0; \mu) = (\pi_\mu^* d_1)(\hat{q}; \mu). \tag{28}$$

One may note that (28) is precisely (13)–(14) expressed with pullbacks through $\pi_{\boldsymbol{\mu}}^*$ and parameter-dependent forms.

Remark 1. It is to be noted that the lift $\hat{\boldsymbol{\ell}}$ in (26) is constructed on the reference domain, and then transported via $\pi_{\boldsymbol{\mu}}^{\mathrm{V}}$ to the parametrized domain. In this construction it is important that $\pi_{\boldsymbol{\mu}}^{\mathrm{V}}\hat{\boldsymbol{\ell}}$ and $\boldsymbol{\ell}$ are compatible, so that the trace of $\boldsymbol{\ell} = \pi_{\boldsymbol{\mu}}^{\mathrm{V}}\hat{\boldsymbol{\ell}}$ on Γ_{D} coincides with the prescribed Dirichlet data.

2.4. Affine representation

We shall assume, and this is critical to the success of what follows, that the forms in (27)–(28) may be expressed as linear combinations of forms that are parameter-*in*dependent.

$$(\pi_\mu^* a)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \mu) = \sum_{i=1}^{N_a} \theta_i^a(\mu)\hat{a}_i(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}), \tag{29}$$

$$(\pi_\mu^* b)(\hat{p}, \hat{\boldsymbol{w}}; \mu) = \sum_{i=1}^{N_b} \theta_i^b(\mu)\hat{b}_i(\hat{p}, \hat{\boldsymbol{w}}), \tag{30}$$

$$(\pi_\mu^* c)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}; \mu) = \sum_{i=1}^{N_c} \theta_i^c(\mu)\hat{c}_i(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}), \tag{31}$$

$$(\pi_\mu^* c_0)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \mu) = \sum_{i=1}^{N_c} \theta_i^{c_0}(\mu)\hat{c}_i^0(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}), \tag{32}$$

$$(\pi_\mu^* d_0)(\hat{\boldsymbol{w}}; \mu) = \sum_{i=1}^{N_1} \theta_i^{d_0}(\mu)\hat{d}_i^0(\hat{\boldsymbol{w}}), \tag{33}$$

$$(\pi_\mu^* d_1)(\hat{q}; \mu) = \sum_{i=1}^{N_2} \theta_i^{d_1}(\mu)\hat{d}_i^1(\hat{q}) \tag{34}$$

Most boundary-condition parameters naturally produce such representations so long as the conditions themselves adhere to the same form, i.e. if $\boldsymbol{g} = \sum_i \xi_i(\mu)\boldsymbol{g}_i$ then we can also write $\hat{\boldsymbol{\ell}} = \sum_i \xi_i(\mu)\hat{\boldsymbol{\ell}}_i$ where $\pi_\mu^{\mathrm{V}}\hat{\boldsymbol{\ell}}_i = \boldsymbol{g}_i$ on Γ_{D}. This expression is substituted into (26), and the resulting affine representation follows directly. The same derivation can be done for Neumann data $\boldsymbol{h}$ and physical parameters such as ν.

Geometric parameters can lead to significantly more complicated representations unless they are of the trivial sort. In this case one must map the integrals (12)–(11) to the reference configuration, and the resulting expressions must be brought into forms compatible with (29)–(34). Since these expressions usually involve the Jacobian and its inverse, it is desirable that they are as simple as possible.

3. Model order reduction

3.1. High-fidelity approximation

The reduced basis is built from underlying high-fidelity finite-element or isogeometric [43] approximations of (27)–(28). To provide a setting for the RB, we therefore first introduce the high-fidelity setting. We introduce a partition $\mathcal{T}^h$ of the reference domain $\hat{\Omega}$ into non-overlapping element domains. The superscript h indicates dependence on a mesh-resolution parameter. We insist that $\mathcal{T}^h$ satisfies the usual uniformity and regularity conditions. The mesh $\mathcal{T}^h$ serves as a support structure for conforming finite-dimensional approximation spaces $\hat{U}_0^h \subset \hat{U}_0$ and $\hat{P}^h \subset \hat{P}$. We assume that the pairing of the velocity and pressure spaces is stable in the inf–sup sense; see, for instance, [41,46–48] and [34,36,38] for stable velocity–pressure pairs in the finite-element and isogeometric setting, respectively. For each $\boldsymbol{\mu} \in \mathcal{P}$, the high-fidelity approximation $(\hat{\boldsymbol{u}}_0^h(\boldsymbol{\mu}), \hat{p}^h(\boldsymbol{\mu}))$ corresponds to the solution of (27)–(28) with $\hat{U}_0$, $\hat{P}$ replaced by their discrete counterparts $\hat{U}_0^h$, $\hat{P}^h$.

The considered high-fidelity approximation spaces $\hat{U}_0^h$, $\hat{P}^h$ must be sufficiently fine to ensure that the error in the high-fidelity approximations is acceptable in accordance with a prescribed tolerance for all considered parameter values $\boldsymbol{\mu} \in \mathcal{P}$. For instance, one may insist that

$$\sup_{\boldsymbol{\mu}\in\mathcal{P}} \left\| \hat{u}_0(\boldsymbol{\mu}) - \hat{u}_0^h(\boldsymbol{\mu}) \right\|_{H^1(\hat{\Omega})} \leq \mathrm{TOL}_0, \qquad \sup_{\boldsymbol{\mu}\in\mathcal{P}} \left\| \hat{p}(\boldsymbol{\mu}) - \hat{p}^h(\boldsymbol{\mu}) \right\|_{L^2(\hat{\Omega})} \leq \mathrm{TOL}_1, \tag{35}$$

for certain prescribed tolerances $\mathrm{TOL}_{0,1}$. The requirements imposed by (35) generally lead to very high dimensionality of $\hat{U}_0^h \times \hat{P}^h$, impeding direct use of the high-fidelity model for fast- or many-query applications.

3.2. Construction of the reduced basis by proper orthogonal decomposition

The main notion underlying RB methods is that in many applications, the dimension of the high-fidelity discrete solution space far exceeds the natural dimension of the parametrized model itself. In other words, it is anticipated that the high-fidelity solution space

$$\left\{ (\hat{\boldsymbol{u}}_0^h(\mu), \hat{p}^h(\mu)) \mid \mu \in \mathcal{P} \right\} \subset \hat{U}_0^h \times \hat{P}^h \tag{36}$$

admits an adequately accurate approximation by a finite-dimensional space whose dimension is significantly lower than that of the high-fidelity approximation space $\hat{U}_0^h \times \hat{P}^h$. This dimension may be considerably more manageable than the dimension of any discretization of $\hat{U}_0 \times \hat{P}$ necessary to achieve sufficient accuracy with conventional methods.

We use standard Proper Orthogonal Decomposition (POD) [19,49] to produce a low-dimensional approximation of the space (36). POD proceeds by sampling $\mathcal{P}$ and generating an *ensemble* of high-fidelity approximations

$$\Phi = \left\{ \varphi_i = (\hat{\boldsymbol{u}}_0^h(\mu_i), \hat{p}^h(\mu_i)) \right\}_{i=1}^N \tag{37}$$

with conventional finite element or isogeometric methods. A covariance matrix is then constructed,

$$C_{ij} = z(\varphi_i, \varphi_j) \tag{38}$$

where the bilinear symmetric positive definite covariance function $z(\cdot,\cdot)$ is left unspecified for the time being. Given eigenpairs $(\lambda_j, \boldsymbol{v}^j)$ $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N)$ of $\boldsymbol{C}$, one may then compose a suitably small number M of basis functions [19, (6.10)]:

$$\psi_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^N v_i^j \varphi_i, \qquad 1 \leq j \leq M. \tag{39}$$

The basis $V = \{\psi_j\}_{j=1}^M$ constitutes a reduced basis in the sense that span $V \subset$ span Φ. The basis V is orthonormal in the $z(\cdot,\cdot)$ inner product by construction and it is optimal in the following sense: Let $W = \{w_j\}_{j=1}^M$ be any orthonormal basis of size M, and $\boldsymbol{P}_W : \Phi \to W$ be the orthogonal projection onto W, i.e.

$$\boldsymbol{P}_W \varphi = \sum_{j=1}^{M} z(\varphi, w_j) w_j \tag{40}$$

Then it holds (see [19, Proposition 6.2])

$$\sum_{i=1}^{N} \|\varphi_i - \boldsymbol{P}_W \varphi_i\|_z^2 \geq \sum_{i=1}^{N} \|\varphi_i - \boldsymbol{P}_V \varphi_i\|_z^2 = \sum_{k=M+1}^{N} \lambda_k. \tag{41}$$

Indeed, since we have by construction that

$$\sum_{i=1}^{N} \|\varphi_i\|_z^2 = \sum_{k=1}^{N} \lambda_k, \tag{42}$$

we obtain the relationship

$$\frac{\sum_{i=1}^N \|\varphi_i - \boldsymbol{P}_V \varphi_i\|_z^2}{\sum_{i=1}^N \|\varphi_i\|_z^2} = \frac{\sum_{k=M+1}^N \lambda_k}{\sum_{k=1}^N \lambda_k}. \tag{43}$$

From this it follows that if $M = M(\epsilon)$ is chosen according to the inequality

$$\sum_{j=1}^{M} \lambda_j \geq \left(1 - \epsilon^2\right) \sum_{j=1}^{N} \lambda_j, \tag{44}$$

then the reduced basis V satisfies the error bound

$$\left(\sum_{i=1}^{N} \left\| \varphi_i - \boldsymbol{P}_V \varphi_i \right\|_z^2 \right)^{1/2} \leq \epsilon \left(\sum_{i=1}^{N} \left\| \varphi_i \right\|_z^2 \right)^{1/2}. \tag{45}$$

If the ensemble Φ is representative of the complete high-fidelity solution space (36) in the sense that the latter is contained in span Φ, then (45) implies the following bound for the projection error:

$$\frac{\left(\int_{\mathcal{P}} \left\| \varphi(\mu) - \boldsymbol{P}_V \varphi(\mu) \right\|_z^2 \, \mathrm{d}\mu \right)^{1/2}}{\left(\int_{\mathcal{P}} \left\| \varphi(\mu) \right\|_z^2 \, \mathrm{d}\mu \right)^{1/2}} \leq \epsilon. \tag{46}$$

3.3. Field separation

In practice, the extraction of a reduced basis from (37) is performed separately on the velocity and pressure components. In other words, first one selects a velocity covariance function $z = z_{\mathrm{v}}$. Typically,

$$z_{\mathrm{v}}\big((\hat{\boldsymbol{u}}_0, \hat{p}), (\hat{\boldsymbol{v}}_0, \hat{q})\big) = \big(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{v}}_0\big)_{H^1(\hat{\Omega})} = \int_{\hat{\Omega}} \nabla \hat{\boldsymbol{u}}_0 : \nabla \hat{\boldsymbol{v}}_0. \tag{47}$$

This is equivalent to using the H^1-seminorm for the velocity space. By virtue of the standing assumption $|\hat{\Gamma}_{\mathrm{D}}| > 0$, this covariance function is symmetric positive definite on U_0. On the product space $U_0 \times P$, it is only semidefinite. The covariance function (47) yields the velocity covariance matrix:

$$C_{ij}^{\mathrm{v}} = \left(\hat{\boldsymbol{u}}_0^h(\boldsymbol{\mu}_i), \hat{\boldsymbol{u}}_0^h(\boldsymbol{\mu}_j) \right)_{H^1(\hat{\Omega})} \tag{48}$$

whose eigenpairs $(\lambda_j, \boldsymbol{v}^j)$ give the reduced velocity basis functions

$$\hat{\boldsymbol{u}}_j^{\mathrm{R}} = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{N} v_i^j \hat{\boldsymbol{u}}_0^h(\boldsymbol{\mu}_i), \qquad 1 \leq j \leq M_{\mathrm{v}} \tag{49}$$

where M_{V} is the chosen dimension of the reduced velocity space. Even though z_{V} is only semidefinite, since the basis is restricted to only those components on which z_{V} *is* definite, the basis functions remain linearly independent.

Independent of the construction of the reduced velocity bases, a pressure basis can be generated by exactly the same method using, for example, the covariance function $z = z_{\mathrm{P}}$ according to:

$$z_{\mathrm{P}}\big((\hat{\boldsymbol{u}}_0, \hat{p}), (\hat{\boldsymbol{v}}_0, \hat{q})\big) = \big(\hat{p}, \hat{q}\big)_{L^2(\hat{\Omega})}. \tag{50}$$

providing a reduced pressure basis:

$$\hat{p}^{\mathrm{R}}_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{N} v_i^j \, \hat{p}^h(\boldsymbol{\mu}_i), \qquad 1 \leq j \leq M_{\mathrm{P}} \tag{51}$$

where M_{P} is the specified dimension of the reduced velocity space, and $(\lambda_j, \boldsymbol{v}^j)$ now refers to the eigenpairs of the pressure covariance matrix.

3.4. Pressure stabilization

The reduced velocity and pressure spaces, U_0^{R}, P^{R}, do not generally inherit the inf–sup stability of the high-fidelity approximation spaces. Given the reduced spaces, the solution of the reduced model generally involves solving a sequence of linear-algebraic problems of the block form

$$\begin{pmatrix} \boldsymbol{A}_{\mathrm{VV}} & \boldsymbol{B}_{\mathrm{VP}} \\ \boldsymbol{B}_{\mathrm{PV}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_{\mathrm{V}} \\ \boldsymbol{f}_{\mathrm{P}} \end{pmatrix} \tag{52}$$

with $(\boldsymbol{B}_{\mathrm{VP}})_{ij} = \boldsymbol{\pi}^*_{\boldsymbol{\mu}} b(\hat{p}^{\mathrm{R}}_j, \hat{\boldsymbol{u}}^{\mathrm{R}}_i; \boldsymbol{\mu})$. On account of the loss of inf–sup stability of the reduced velocity–pressure pair, one typically observes that the matrix $\boldsymbol{B}_{\mathrm{VP}}$ is row-rank-deficient, i.e. it has a nontrivial kernel, therefore leading to unstable pressure solutions. One procedure to resolve this instability is to artificially enrich the velocity space with so-called *supremizers*, viz. basis functions whose purpose is not to achieve greater approximative power but rather to keep the system well-posed; see [31]. We will use this technique both for comparative purposes and for generating pressure solutions for velocity-only discretizations, see Section 4.1.

For each pressure snapshot $\hat{p}^h(\boldsymbol{\mu}_i)$, the corresponding supremizer is the velocity function in $\hat{U}^h_0$ that realizes the supremum in the LBB[1] condition,

$$\inf_{\hat{p}^h \in \hat{P}^h} \sup_{\hat{\boldsymbol{u}}^h_0 \in \hat{U}^h_0} \frac{(\boldsymbol{\pi}^*_{\boldsymbol{\mu}} b)(\hat{p}^h, \hat{\boldsymbol{u}}^h_0; \mu)}{\|\hat{p}^h\|_{L^2(\hat{\Omega})} \|\hat{\boldsymbol{u}}^h_0\|_{H^1(\hat{\Omega})}} \geq \beta_h > 0; \tag{53}$$

see [31] for the theoretical background. The supremizers can in fact be conceived of as Riesz representations of the pressure solutions as viewed through the $\boldsymbol{\pi}^*_{\boldsymbol{\mu}} b(\cdot, \cdot; \mu)$-form. In other words, given a reduced pressure function $\hat{p}^{\mathrm{R}}$, its corresponding supremizer $\hat{\boldsymbol{s}}^h_0$ is the Riesz representation in $\hat{U}^h_0$ of the linear functional $\boldsymbol{\pi}^*_{\boldsymbol{\mu}} b(\hat{p}^h, \cdot; \mu)$, i.e. the solution of

$$\hat{\boldsymbol{s}}^h_0 \in \hat{U}^h_0 : \qquad \Big(\hat{\boldsymbol{w}}^h_0, \hat{\boldsymbol{s}}^h_0\Big)_{H^1(\hat{\Omega})} = (\boldsymbol{\pi}^*_{\boldsymbol{\mu}} b)(\hat{p}^h, \hat{\boldsymbol{w}}^h_0; \mu) \qquad \forall \hat{\boldsymbol{w}}^h_0 \in \hat{U}^h_0. \tag{54}$$

From (54) it follows that the supremizers are formally $\boldsymbol{\mu}$-dependent functions, via the dependence of the right member of (54) on $\boldsymbol{\mu}$, although this dependence is generally restricted to geometric parameters. They are therefore not directly suitable for use as basis functions in a reduced-basis method. To bypass the $\boldsymbol{\mu}$-dependence of the supremizers, we apply the process described as "approximate supremizer enrichment" in [31, 4.2.3]. In this method, supremizer snapshots $\hat{\boldsymbol{s}}^h_0(\mu_i)$ are constructed offline in tandem with the pressure snapshots $\hat{p}^h(\mu_i)$. These supremizer snapshots are then reduced via POD in exactly the same manner as the velocity snapshots, into a separate and independent reduced space, which is subsequently added to the reduced velocity space via a direct-sum composition. Formally, the sum space should be orthogonalized to ensure linear independence. Following [31], we have ignored this on the presumption that the reduced velocity space and the supremizer space are sufficiently disparate to retain linear independence. This hypothesis is verified in the numerical experiments in Section 6.3.

[1] Ladyzhenskaya–Babuška–Brezzi, also known as the inf–sup condition.

Introduction of the supremizer modes yields the following extension of the linear-algebraic systems (52):

$$\begin{pmatrix} \boldsymbol{A}_{\mathrm{VV}} & \boldsymbol{A}_{\mathrm{VS}} & \boldsymbol{B}_{\mathrm{VP}} \\ \boldsymbol{A}_{\mathrm{SV}} & \boldsymbol{A}_{\mathrm{SS}} & \boldsymbol{B}_{\mathrm{SP}} \\ \boldsymbol{B}_{\mathrm{PV}} & \boldsymbol{B}_{\mathrm{PS}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_{\mathrm{V}} \\ \boldsymbol{f}_{\mathrm{S}} \\ \boldsymbol{f}_{\mathrm{P}} \end{pmatrix} \tag{55}$$

One may note that with *exact* supremizer enrichment, the practical consequence of (54) is that $\boldsymbol{B}_{\mathrm{SP}}$ coincides with the Gramian matrix $(\boldsymbol{B}_{\mathrm{SP}})_{ij} = (\hat{\boldsymbol{s}}_i^h, \hat{\boldsymbol{s}}_j^h)_{H^1(\hat{\Omega})}$, which is nonsingular by design as discussed in Section 3.

It must be pointed out that the purpose of supremizers is to serve as a *test space* and that they do not meaningfully contribute to the approximative power of the method. The coefficient vector $\boldsymbol{s}$ makes no significant contribution to the solution. In Section 4.1 we shall see that they even vanish identically in the divergence-conforming case.

Given pressure and velocity spaces, a simple procedure is outlined in [31] for checking (53) numerically by way of solving a generalized eigenvalue problem,

$$\beta_h(\mu) = \sqrt{\lambda_{\min}(\mu)}, \tag{56}$$

where $\lambda_{\min}(\mu)$ is the minimal eigenvalue solution of

$$\begin{pmatrix} \boldsymbol{B}_{\mathrm{PV}} & \boldsymbol{B}_{\mathrm{PS}} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}_{\mathrm{VV}} & \boldsymbol{M}_{\mathrm{VS}} \\ \boldsymbol{M}_{\mathrm{SV}} & \boldsymbol{M}_{\mathrm{SS}} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{B}_{\mathrm{VP}} \\ \boldsymbol{B}_{\mathrm{SP}} \end{pmatrix} \boldsymbol{p} = \lambda \boldsymbol{M}_{pp} \boldsymbol{p}, \tag{57}$$

$\boldsymbol{M}$ denoting the H^1-seminorm mass matrix or the L^2-mass matrix, respectively.

The computation of supremizers introduces an additional offline cost which in our experience is quite negligible. The solution of (54) for every parameter sample is cheap next to the more challenging Navier–Stokes problem. A cost is also paid in the online stage, as the system (55) is larger than (52) by several degrees of freedom that do not meaningfully contribute to better approximations, as we will see shortly.

It must be noted that the supremizer method is not the only stabilization method proposed in the literature, see for example [50,51], where reduced basis methods are stabilized based on fine-scale Variational Multiscale Methods (VMS).

3.5. Assembly of the reduced system

Let $\boldsymbol{V} \in \mathbb{R}^{N \times M}$ be a μ-independent tall *transformation matrix*, mapping coefficient vectors in the reduced basis to coefficient vectors in the original high-fidelity basis, i.e. the columns of $\boldsymbol{V}$ are the high-fidelity coefficient vectors of the reduced basis functions.

We aim then to express the solution to a generic finite element system $\boldsymbol{A}(\mu)\boldsymbol{u}(\mu) = \boldsymbol{f}(\mu)$ by the reduced coefficient vector $\boldsymbol{u}(\mu) = \boldsymbol{V}\boldsymbol{u}^{\mathrm{R}}(\mu)$, i.e.

$$\boldsymbol{A}(\mu)\boldsymbol{V}\boldsymbol{u}^{\mathrm{R}}(\mu) = \boldsymbol{f}(\mu). \tag{58}$$

Since (58) is overdetermined, we choose the Galerkin approach of enforcing that the residual must be orthogonal to the reduced space, i.e.

$$\boldsymbol{V}^{\intercal}\left(\boldsymbol{A}(\mu)\boldsymbol{V}\boldsymbol{u}^{\mathrm{R}}(\mu) - \boldsymbol{f}(\mu)\right) = \boldsymbol{0}, \tag{59}$$

or, more traditionally,

$$\boldsymbol{V}^{\intercal}\boldsymbol{A}(\mu)\boldsymbol{V}\boldsymbol{u}^{\mathrm{R}}(\mu) = \boldsymbol{V}^{\intercal}\boldsymbol{f}(\mu). \tag{60}$$

The importance of (29)–(34) then becomes clear. Although the solution of the reduced problem involves nonlinear systems of size M, which are quick to solve when M is small (even if they are dense, as they will be), the assembly of these systems remains nontrivial. However, given (60), each matrix and vector corresponding to a bilinear or linear form in (29)–(34) may be reduced to the new basis independently without regard to parametric dependence. It is then vital that $\boldsymbol{V}$ is indeed parameter-independent.

For example, let us consider (29). Let the matrix $\boldsymbol{M}_i^{\mathrm{H}}$ be the high-fidelity matrix corresponding to the bilinear form $\hat{a}_i$. (Here, superscript H refers to *high-fidelity* and superscript R refers to the reduced model.) Then we have

$$\boldsymbol{M}^{\mathrm{R}}(\mu) = \boldsymbol{V}^{\intercal}\boldsymbol{M}^{\mathrm{H}}(\mu)\boldsymbol{V} = \boldsymbol{V}^{\intercal}\left(\sum_i \theta_i^d(\mu)\boldsymbol{M}_i^{\mathrm{H}}\right)\boldsymbol{V} = \sum_i \theta_i^d(\mu)\underbrace{\boldsymbol{V}^{\intercal}\boldsymbol{M}_i^{\mathrm{H}}\boldsymbol{V}}_{\boldsymbol{M}_i^{\mathrm{R}}}, \tag{61}$$

where each of the small matrices $\boldsymbol{M}_i^{\mathrm{R}}$ can be computed in the offline stage. Assembling the reduced model system matrices then becomes trivial.

3.6. Assembly of nonlinear terms

Typically, the presence of a nonlinear term such as $(\pi_\mu^* c)(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{u}}_0, \hat{\boldsymbol{w}}; \mu)$ in (27) is treated by introducing a nonlinear solver (e.g. Newton's method or Picard fixed-point iteration) that is based on solving a sequence of *linear* systems. These linear systems generally involve (an approximation to) the Fréchet derivative of the nonlinear term, evaluated at a sequence of approximations to the solution of the nonlinear problem. Because the sequence of approximations is evidently not known in advance, the assembly of the Jacobian matrices corresponding to the Fréchet derivatives is generally done by a standard assembly procedure that includes an integration loop. However, the computational cost of such an assembly operation is prohibitive in the online stage of an RBM. Currently, there is no standard methodology for the efficient treatment of general nonlinear terms in the online stage of an RBM.

The nonlinear term in the Navier–Stokes equations has a particular structure that can be exploited to obtain an efficient assembly in the online stage. Specifically, the nonlinear term in (27) is quadratically nonlinear, as indicated by the fact that it can be represented as a trilinear form with a repeated argument. The quadrature-based assembly of the Jacobian matrix in the online stage can then be avoided by assembling in the offline stage the RB third-order tensors corresponding to the components $\hat{c}_i$ in the affine representation (31) of the trilinear form. In particular, one can construct the third-order tensors $\{\boldsymbol{C}^i\}_{i=1}^{N_c}$ with components:

$$C^i_{jkl} = \hat{c}_i\big(\hat{\boldsymbol{u}}_j^{\mathrm{R}}, \hat{\boldsymbol{u}}_k^{\mathrm{R}}, \hat{\boldsymbol{u}}_l^{\mathrm{R}}\big) \qquad j,k,l \in \{1,2,\dots,M_{\mathrm{V}}\} \tag{62}$$

For any approximation $\underline{\hat{\boldsymbol{u}}}_0 \in U_0^{\mathrm{R}}$ the Jacobian matrix associated with the Fréchet derivative of $\pi_\mu^* c$ restricted to the reduced velocity space U_0^{R} can then be assembled in accordance with the ultimate expression in the chain of identities:

$$\begin{aligned}\frac{d}{ds}(\pi_\mu^* c)(\underline{\hat{\boldsymbol{u}}}_0 + s\hat{\boldsymbol{u}}_j^{\mathrm{R}}, \underline{\hat{\boldsymbol{u}}}_0 + s\hat{\boldsymbol{u}}_j^{\mathrm{R}}, \hat{\boldsymbol{u}}_k^{\mathrm{R}}; \mu)\Big|_{s=0} &= (\pi_\mu^* c)(\underline{\hat{\boldsymbol{u}}}_0, \hat{\boldsymbol{u}}_j^{\mathrm{R}}, \hat{\boldsymbol{u}}_k^{\mathrm{R}}; \mu) + (\pi_\mu^* c)(\hat{\boldsymbol{u}}_j^{\mathrm{R}}, \underline{\hat{\boldsymbol{u}}}_0, \hat{\boldsymbol{u}}_k^{\mathrm{R}}; \mu)\\ &= \sum_{i=1}^{N_c} \theta_i^c(\mu) \sum_{l=1}^{M_{\mathrm{V}}} \big(C^i_{ljk} + C^i_{jlk}\big)\eta_l\end{aligned} \tag{63}$$

where η_l denote the coefficients of $\underline{\hat{\boldsymbol{u}}}_0$ relative to the reduced basis, i.e. $\underline{\hat{\boldsymbol{u}}}_0 = \sum_{l=1}^{M_{\mathrm{V}}} \eta_l \hat{\boldsymbol{u}}_l^{\mathrm{R}}$. Hence, for any approximation $\underline{\hat{\boldsymbol{u}}}_0 \in U_0^{\mathrm{R}}$, the Jacobian matrix can be assembled on the basis of the pre-computed third-order tensors. For large models, the storage requirements for the third-order tensor are unmanageable. However, the storage of such a tensor for a reduced space *is* feasible. Although these tensors are dense, we find that the modest sizes keep the storage requirements realistic, typically in the regions of hundreds of megabytes for ∼100 basis functions.

4. Divergence-conforming methods

4.1. Separated velocity and pressure bases

In high-fidelity terms, a divergence-conforming method is any method which ensures that the pressure space is exactly equal to the divergence of the velocity space. Today this sort of method is often seen in the field of isogeometric analysis, where the use of function spaces based on B-splines, together with the Piola transformation, creates sufficient flexibility to easily work with such spaces.

The practical consequence of such a method is that it eliminates all forms of pressure instabilities and simultaneously guarantees both a stable method for convection-dominated flow problems and a velocity solution that is pointwise solenoidal (divergence-free). The latter point is interesting, since it paves the way to generate fully divergence-free reduced velocity spaces.

To ensure that solenoidal velocity fields remain solenoidal even in the case of parameter-dependent geometries, one employs the Piola transform, whereby the physical manifestation of the reference velocity field $\hat{\boldsymbol{u}}$ is not the canonical transformation (25) (which, indeed, does not preserve the solenoidal property in general), but rather

$$\boldsymbol{u} = \frac{\boldsymbol{J}}{|\boldsymbol{J}|}(\hat{\boldsymbol{u}} \circ \chi_\mu) \tag{64}$$

where $\boldsymbol{J}$ is the Jacobian of the mapping χ_{μ}^{-1}. This modified mapping must then be used in (27)–(28). It should be noted that the affine representations (29)–(34) may be considerably more complicated in this case, since $\boldsymbol{J}$ in general is parameter-dependent.

We must also point out that to ensure the success of this method, the lift functions $\hat{\boldsymbol{\ell}}(\mu)$ *must* also be solenoidal. Since the total solution $\hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}_0 + \hat{\boldsymbol{\ell}}$, and since the reduced spaces are all concerned with the approximation of $\hat{\boldsymbol{u}}_0$ (and not $\hat{\boldsymbol{u}}$), this is necessary to ensure the equivalence between the solenoidality of $\hat{\boldsymbol{u}}_0$ and that of $\hat{\boldsymbol{u}}$. In practice one may solve the Stokes problem with a divergence-conforming method to generate solenoidal lift functions. Since the lift is constructed in the offline stage, the cost of this construction can be ignored.

With a fully solenoidal reduced velocity space, we are free to employ the velocity-only formulation (18), so that (55) reduces to

$$\boldsymbol{A}_{\mathrm{VV}}\boldsymbol{u} = \boldsymbol{f}_{\mathrm{V}}. \tag{65}$$

which is a system of size M compared to (55) of size $3M$.

The supremizers of Section 3.4 function as the test functions in the pressure recovery equation (19). Indeed, one obtains the system

$$\boldsymbol{B}_{\mathrm{SP}}\boldsymbol{p} = \boldsymbol{f}_{\mathrm{S}} - \boldsymbol{A}_{\mathrm{SV}}\boldsymbol{u}. \tag{66}$$

It is worth noting that the same expressions can be reached by setting $\boldsymbol{B}_{\mathrm{VP}} = 0$ and $\boldsymbol{f}_{\mathrm{P}} = 0$ in (55), obtaining

$$\begin{pmatrix} \boldsymbol{A}_{\mathrm{VV}} & \boldsymbol{A}_{\mathrm{VS}} & \\ \boldsymbol{A}_{\mathrm{SV}} & \boldsymbol{A}_{\mathrm{SS}} & \boldsymbol{B}_{\mathrm{SP}} \\ & \boldsymbol{B}_{\mathrm{PS}} & \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{s} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}_{\mathrm{V}} \\ \boldsymbol{r}_{\mathrm{S}} \\ \boldsymbol{0} \end{pmatrix} \tag{67}$$

If all spaces are of equal size M, the system (67) is block-triangular, and can be solved as (65)–(66) together with $\boldsymbol{s} = 0$. This illustrates the fact that supremizers serve no approximative purpose and merely function as a test space for the pressure.

The system (65)–(66) also facilitates a much faster reduced method than (55), solving two systems of size M rather than one of size $3M$. We shall see in the following that this may lead to speedups of more than an order of magnitude. It is also worth pointing out that since reduced systems are dense rather than sparse, the benefits of this reduction in size is amplified further compared to the usual situation of high-fidelity sparse systems.

For other methods for pressure stabilization based on pure velocity solutions followed by pressure recovery in various forms, see [52–54]. Other methods for pressure recovery from pure velocity solutions have been investigated. In [53], three different methods based on modified equations and spaces are compared. The authors in [52] compute the pressure by taking the divergence of the Navier–Stokes equation, projecting the result onto the pressure modes.

4.2. Combined velocity–pressure basis

One interesting method of pressure reconstruction is given in [54], where all reduced basis functions are *joint* velocity *and* pressure functions. The pressure is then reconstructed from the coefficients of the solution vector in the reduced space, exactly as with the velocity field. In the following, we will refer to this technique as *pressure reconstruction*, to distinguish it from *pressure recovery* as discussed in Section 4.1.

It is tempting to introduce this approach to divergence-conforming RBMs because the system (65) remains the same, that is, adding pressure data to divergence-free velocity basis functions does not change the reduced system.

Remark 2. This method implies a linear dependence of pressure on velocity, an assumption that contradicts the Navier–Stokes equations. Its application should be restricted to linear models (e.g. Stokes flow) or problems where the parameter space is suitably small for the linearization to be valid.

There are at least two ways to construct a combined basis. One way is to omit the separation discussed in Section 3.3, and instead pick z in (38) as joint velocity–pressure inner product, for example

$$z\big((\hat{\boldsymbol{u}}_0, \hat{p}), (\hat{\boldsymbol{v}}_0, \hat{q})\big) = \big(\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{v}}_0\big)_{H^1(\hat{\Omega})} + \rho\,\big(\hat{p}, \hat{q}\big)_{L^2(\hat{\Omega})} \tag{68}$$

where $\rho > 0$ can be chosen to appropriately scale the relative importance of velocity and pressure. The POD procedure will then produce joint velocity–pressure basis functions whose velocity parts are divergence-free, so long as the snapshots are also divergence-free.

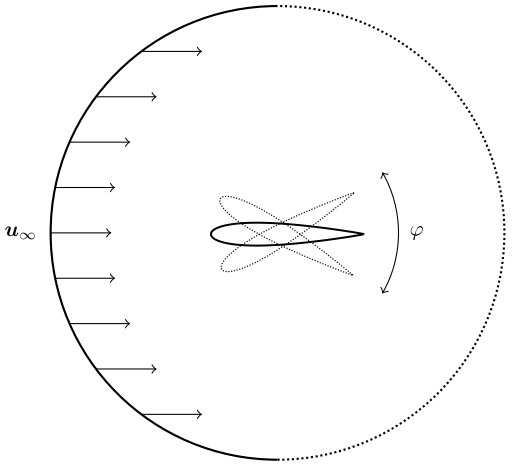

Fig. 1. Sketch of the airfoil flow problem and its parameters. Solid lines indicate Dirichlet boundaries, and dotted lines indicate Neumann boundaries.

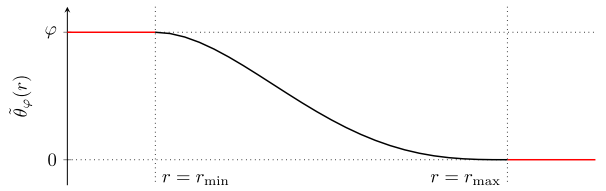

Fig. 2. $\tilde{\theta}_\varphi(r)$ as a function of r.

Another approach is to construct the velocity basis according to Section 3.3, and then to use the same coefficients v_i^j as in (49) to generate the pressure functions in (51). Note that this does not guarantee any kind of linear independence in the pressure component.

With (68) one obtains combined velocity–pressure basis functions that are orthonormal in a joint inner product. The second method produces the same velocity basis as the velocity-only formulation discussed in Section 4.1 with attached pressure data that may or may not be linearly independent. However, because pressure reconstruction is based on direct linear combination of the pressure modes, linear independence of these modes is not required.

In the following we have used the second approach for pressure reconstruction.

5. Numerical experiments

We consider a two-dimensional airfoil suspended in a mesh at the origin. We wish to perform airflow simulations around this airfoil with the angle of attack φ and the inflow velocity u_∞ as parameters, in other words $\mu = (\varphi, u_\infty)$; See Fig. 1 for a sketch. The purpose of the present section is to develop an affine representation of this problem, as understood by (29)–(34).

While the velocity is trivial, the geometric parameter can be accomplished by rotating the mesh through an angle that depends on the distance from the center of the airfoil (hereafter referred to as the *radius* r), i.e. $\tilde{\theta}_\varphi = \tilde{\theta}_\varphi(r)$ so that

$$\tilde{\theta}_\varphi(r) = \begin{cases} \varphi, & r \leq r_{\min}, \\ 0, & r \geq r_{\max}, \end{cases}$$

and which smoothly interpolates between the two in the region $r_{\min} < r < r_{\max}$. See Fig. 2 for an example. Note that we also require

$$\tilde{\theta}'_\varphi(r_{\min}) = \tilde{\theta}'_\varphi(r_{\max}) = 0$$

in accordance with Remark 1, i.e. the mapping of lifts between reference and transformed domains will not modify the boundary data. It is desirable that the entire airfoil is contained within the region $r < r_{\min}$, and that the boundary of the mesh is entirely inside the region $r > r_{\max}$. Given these restrictions, the mesh may otherwise be freely chosen.

In the following we will denote by $\theta(r)$ a *canonical* angle function, satisfying

$$\theta(r) = \begin{cases} 1, & r \leq r_{\min}, \\ 0, & r \geq r_{\max}. \end{cases}$$

With this, we can write $\tilde{\theta}_\varphi(r) = \varphi\theta(r)$. The mapping from reference coordinates to physical coordinates can then be expressed as

$$\chi_\varphi^{-1}\hat{\boldsymbol{r}} = \boldsymbol{R}(\varphi\theta(r))\,\hat{\boldsymbol{r}},$$

where $\hat{\boldsymbol{r}} = (\hat{x}, \hat{y})$, $r = \|\hat{\boldsymbol{r}}\| = \sqrt{\hat{x}^2 + \hat{y}^2}$ and $\boldsymbol{R}$ is the rotation matrix,

$$\boldsymbol{R}(a) = \begin{pmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{pmatrix}.$$

The Jacobian $\boldsymbol{J} = \boldsymbol{J}(\chi_\varphi^{-1})$ can be expressed, using $a = \varphi\theta(r)$ as shorthand, as

$$\begin{aligned} \boldsymbol{J} &= \boldsymbol{R}(a) + \boldsymbol{R}'(a)\hat{\boldsymbol{r}}\nabla a^\intercal = \boldsymbol{R}(a)\left(\boldsymbol{I} + \boldsymbol{P}\hat{\boldsymbol{r}}\nabla a^\intercal\right) \\ &= \boldsymbol{R}(a)\left(\boldsymbol{I} + \frac{\varphi\theta'}{r}\boldsymbol{P}\hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^\intercal\right) = \boldsymbol{R}(a)\left(\boldsymbol{I} + \varphi\boldsymbol{P}\boldsymbol{Q}\right) \end{aligned} \tag{69}$$

where $\boldsymbol{R}'(a) = \partial\boldsymbol{R}(a)/\partial a$ and where the two utility matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are defined as

$$\boldsymbol{P} = \boldsymbol{R}(\pi/2), \qquad \boldsymbol{Q} = \frac{\theta'}{r}\hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^\intercal, \tag{70}$$

noting the useful property that $\boldsymbol{R}\boldsymbol{P} = \boldsymbol{R}'$.

The determinant of $\boldsymbol{R}$ is 1 and the determinant of $\boldsymbol{J}$ follows from the matrix determinant lemma (as a one-rank update to an invertible matrix),

$$\det\boldsymbol{J} = 1 + \frac{\varphi\theta'}{r}\hat{\boldsymbol{r}}^\intercal\boldsymbol{P}\hat{\boldsymbol{r}} = 1,$$

since the last term may be recognized as the inner product between two orthogonal vectors. The inverse of $\boldsymbol{J}$ follows from the Sherman–Morrison theorem,

$$\boldsymbol{J}^{-1} = (\boldsymbol{I} - \varphi\boldsymbol{P}\boldsymbol{Q})\,\boldsymbol{R}(a)^\intercal.$$

It can then be seen that the problem of finding affine representations of $\boldsymbol{J}$ and $\boldsymbol{J}^{-1}$ reduces to the affine representation of $\boldsymbol{R}$. We therefore proceed as follows.

$$\boldsymbol{R}(a) = \begin{pmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{pmatrix} = \sum_{i=0}^{\infty}\left(\frac{(-1)^i a^{2i}}{(2i)!}\boldsymbol{I} + \frac{(-1)^i a^{2i+1}}{(2i+1)!}\boldsymbol{P}\right).$$

Noting that, since $\boldsymbol{P}^0 = \boldsymbol{I}, \boldsymbol{P}^1 = \boldsymbol{P}, \boldsymbol{P}^2 = -\boldsymbol{I}, \boldsymbol{P}^3 = -\boldsymbol{P}$ etc., we obtain

$$(-1)^i\boldsymbol{I} = \boldsymbol{P}^{2i}, \qquad (-1)^i\boldsymbol{P} = \boldsymbol{P}^{2i+1},$$

so the series expansion can be more succinctly written as

$$\boldsymbol{R}(a) = \sum_{i=0}^{\infty}\frac{a^i}{i!}\boldsymbol{P}^i = \sum_{i=0}^{\infty}\varphi^i\underbrace{\frac{\theta^i}{i!}\boldsymbol{P}^i}_{\boldsymbol{R}_i} = \sum_{i=0}^{\infty}\varphi^i\boldsymbol{R}_i. \tag{71}$$

To investigate the effect of the Piola transformation, we consider the affine representations (29)–(32) for two different methods: first, a conventional method using the pullback via $\pi_{\boldsymbol{\mu}}^{\mathrm{v}}$ to map between function spaces, and second a divergence-conforming method based on the Piola mapping (64). In doing so we will truncate the series for $\boldsymbol{R}(a)$ to a number of terms that can achieve the desired precision. It is important to note that the system (67) only obtains its desired form for high accuracy approximations, and that if the affine representation form for $b(\cdot, \cdot, \mu)$ is not exact, the

matrix $\boldsymbol{B}_{\text{VP}}$ from (55) will correspondingly not be exactly zero, and the solution of (65)–(66) will not agree with the solution of (55).

The error made by truncating (71) to n terms is approximately $\varphi_{\text{MAX}}^n/n!$. For a maximal angle of attack of 35° for example, we can expect about 10 digits of accuracy with $n = 10$ terms.

In the following, only the parameter φ is considered, as u_∞ is comparatively trivial. In all transformations, the Jacobian determinant $|\boldsymbol{J}|$ is identically equal to 1.

In Section 5.1 we will derive the non-Piola formulation, and in Section 5.2 we will make the same derivations for the Piola formulations. It will be seen that the latter, while simplified somewhat due to the precise nature of $\boldsymbol{J}$ in the present case, is nevertheless more complicated than the former. For more general geometric variations (e.g. a deforming airfoil) it is likely that this explicit approach would have to be abandoned for an automatic black-box type algorithm such as Empirical Interpolation (EIM, see [55–57] and [19, Chapter 10]).

5.1. Non-Piola formulation

For the Laplacian form a we get

$$(\pi_\mu^* a)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \varphi) = \nu \int_{\hat{\Omega}} (\boldsymbol{J}^{-\mathsf{T}}\nabla)\hat{\boldsymbol{u}} : (\boldsymbol{J}^{-\mathsf{T}}\nabla)\hat{\boldsymbol{w}}|\boldsymbol{J}| = \nu \int_{\hat{\Omega}} \nabla\hat{\boldsymbol{u}} : (\boldsymbol{J}^{-1}\boldsymbol{J}^{-\mathsf{T}}\nabla)\hat{\boldsymbol{w}}|\boldsymbol{J}|, \tag{72}$$

and we find for the matrix $\boldsymbol{J}^{-1}\boldsymbol{J}^{-\mathsf{T}}$ that

$$\begin{aligned}\boldsymbol{J}^{-1}\boldsymbol{J}^{-\mathsf{T}} &= (\boldsymbol{I} - \varphi\boldsymbol{P}\boldsymbol{Q})\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}(\boldsymbol{I} + \varphi\boldsymbol{Q}\boldsymbol{P}) \\ &= \boldsymbol{I} + \varphi\underbrace{(\boldsymbol{Q}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{Q})}_{\boldsymbol{D}_1} - \varphi^2\underbrace{\boldsymbol{P}\boldsymbol{Q}^2\boldsymbol{P}}_{\boldsymbol{D}_2} \\ &= \boldsymbol{I} + \varphi\boldsymbol{D}_1 - \varphi^2\boldsymbol{D}_2.\end{aligned} \tag{73}$$

Since $|\boldsymbol{J}| = 1$, this gives a three-term affine representation of a as

$$(\pi_\mu^* a)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{w}}; \varphi) = \nu \int_{\hat{\Omega}} \nabla\hat{\boldsymbol{u}} : \nabla\hat{\boldsymbol{w}} + \nu\varphi \int_{\hat{\Omega}} \nabla\hat{\boldsymbol{u}} : (\boldsymbol{D}_1\nabla)\hat{\boldsymbol{w}} - \nu\varphi^2 \int_{\hat{\Omega}} \nabla\hat{\boldsymbol{u}} : (\boldsymbol{D}_2\nabla)\hat{\boldsymbol{w}}. \tag{74}$$

For the divergence form b we have

$$(\pi_\mu^* b)(\hat{p}, \hat{\boldsymbol{w}}; \varphi) = \int_{\hat{\Omega}} \hat{p}(\boldsymbol{J}^{-\mathsf{T}}\nabla) \cdot \hat{\boldsymbol{w}}|\boldsymbol{J}| = \int_{\hat{\Omega}} \hat{p}\boldsymbol{J}^{-\mathsf{T}} : \nabla\hat{\boldsymbol{w}}|\boldsymbol{J}|, \tag{75}$$

meaning we need a series representation of $\boldsymbol{J}^{-\mathsf{T}}$:

$$\begin{aligned}\boldsymbol{J}^{-\mathsf{T}} &= \boldsymbol{R}(a)(\boldsymbol{I} - \varphi\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{P}^{\mathsf{T}}) = \boldsymbol{R}(a)(\boldsymbol{I} + \varphi\boldsymbol{Q}\boldsymbol{P}) = \sum_{i=0}^{\infty} \varphi^i \boldsymbol{R}_i(\boldsymbol{I} + \varphi\boldsymbol{Q}\boldsymbol{P}) \\ &= \sum_{i=0}^{\infty} \varphi^i \boldsymbol{R}_i + \varphi^{i+1}\boldsymbol{R}_i\boldsymbol{Q}\boldsymbol{P} = \sum_{i=0}^{\infty} \varphi^i \underbrace{(\boldsymbol{R}_i + \boldsymbol{R}_{i-1}\boldsymbol{Q}\boldsymbol{P})}_{\boldsymbol{B}_i^{(-)}} = \sum_{i=0}^{\infty} \varphi^i \boldsymbol{B}_i^{(-)},\end{aligned} \tag{76}$$

with the understanding that $\boldsymbol{R}_{-1} = 0$. This gives an affine representation of b in $2n$ terms, where n is a suitable number of terms for a truncated Taylor series for sin or cos, given the range of φ under consideration.

$$(\pi_\mu^* b)(\hat{p}, \hat{\boldsymbol{w}}; \varphi) \approx \sum_{i=0}^{2n} \varphi^i \int_{\hat{\Omega}} \hat{p}\boldsymbol{B}_i^{(-)} : \nabla\hat{\boldsymbol{w}} \tag{77}$$

The same expansion will work with the convective term c, viz.

$$(\pi_\mu^* c)(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}; \varphi) = \int_{\hat{\Omega}} (\hat{\boldsymbol{u}} \cdot \boldsymbol{J}^{-\mathsf{T}}\nabla)\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{w}}|\boldsymbol{J}| \approx \sum_{i=0}^{2n} \varphi^i \int_{\hat{\Omega}} (\hat{\boldsymbol{u}} \cdot \boldsymbol{B}_i^{(-)}\nabla)\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{w}}. \tag{78}$$

5.2. Piola formulation

This proceeds as for the non-Piola case, except every vector field is pre-multiplied with the Jacobian (recall, the determinant is 1, which simplifies (64) somewhat). For this, we need an affine representation of $\boldsymbol{J}$. It looks deceptively like that of $\boldsymbol{J}^{-\mathsf{T}}$:

$$\begin{aligned}\boldsymbol{J} &= \boldsymbol{R}(a)(\boldsymbol{I}+\varphi\boldsymbol{P}\boldsymbol{Q}) = \sum_{i=0}^{\infty}\varphi^i\boldsymbol{R}_i(\boldsymbol{I}+\varphi\boldsymbol{P}\boldsymbol{Q}) = \sum_{i=0}^{\infty}\varphi^i\boldsymbol{R}_i + \varphi^{i+1}\boldsymbol{P}\boldsymbol{Q}\\ &= \sum_{i=0}^{\infty}\varphi^i\underbrace{(\boldsymbol{R}_i + \boldsymbol{R}_{i-1}\boldsymbol{P}\boldsymbol{Q})}_{\boldsymbol{B}_i^{(+)}} = \sum_{i=0}^{\infty}\varphi^i\boldsymbol{B}_i^{(+)}.\end{aligned} \tag{79}$$

For the form b we then get

$$\begin{aligned}(\pi_\mu^* b)(\hat{p},\hat{\boldsymbol{w}};\varphi) &= \int_{\hat{\Omega}}\hat{p}\boldsymbol{J}^{-\mathsf{T}} : \nabla(\boldsymbol{J}\hat{\boldsymbol{w}})|\boldsymbol{J}| \approx \int_{\hat{\Omega}}\hat{p}\left(\sum_{i=0}^{2n}\varphi^i\boldsymbol{B}_i^{(-)}\right) : \nabla\left(\sum_{j=0}^{2n}\varphi^j\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}\right)\\ &= \sum_{i,j=0}^{2n}\varphi^{i+j}\int_{\hat{\Omega}}\hat{p}\boldsymbol{B}_i^{(-)} : \nabla\left(\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}\right).\end{aligned} \tag{80}$$

Given truncated expressions for $\boldsymbol{J}$ and $\boldsymbol{J}^{-\mathsf{T}}$ with $2n$ terms, this is an affine representation with $4n$ terms.

The form c has similar complexity.

$$\begin{aligned}(\pi_\mu^* c)(\hat{\boldsymbol{u}},\hat{\boldsymbol{v}},\hat{\boldsymbol{w}};\varphi) &= \int_{\hat{\Omega}}(\boldsymbol{J}\hat{\boldsymbol{u}}\cdot\boldsymbol{J}^{-\mathsf{T}}\nabla)\boldsymbol{J}\hat{\boldsymbol{v}}\cdot\boldsymbol{J}\hat{\boldsymbol{w}}|\boldsymbol{J}| \approx \int_{\hat{\Omega}}(\hat{\boldsymbol{u}}\cdot\nabla)\left(\sum_{i=0}^{2n}\varphi^i\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{v}}\right)\cdot\left(\sum_{i=0}^{2n}\varphi^j\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}\right)\\ &= \sum_{i,j=0}^{2n}\varphi^{i+j}\int_{\hat{\Omega}}(\hat{\boldsymbol{u}}\cdot\nabla)\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{v}}\cdot\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}.\end{aligned} \tag{81}$$

And finally, a can be represented as

$$\begin{aligned}(\pi_\mu^* a)(\hat{\boldsymbol{u}},\hat{\boldsymbol{w}};\varphi) &= \int_{\hat{\Omega}}(\boldsymbol{J}^{-\mathsf{T}}\nabla)(\boldsymbol{J}\hat{\boldsymbol{u}}) : (\boldsymbol{J}^{-\mathsf{T}}\nabla)(\boldsymbol{J}\hat{\boldsymbol{w}})|\boldsymbol{J}| = \int_{\hat{\Omega}}\nabla(\boldsymbol{J}\hat{\boldsymbol{u}}) : (\boldsymbol{J}^{-1}\boldsymbol{J}^{-\mathsf{T}}\nabla)(\boldsymbol{J}\hat{\boldsymbol{w}})|\boldsymbol{J}|\\ &\approx \int_{\hat{\Omega}}\nabla\left(\sum_{i=0}^{2n}\varphi^i\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{u}}\right) : ((\boldsymbol{I}+\varphi\boldsymbol{D}_1-\varphi^2\boldsymbol{D}_2)\nabla)\left(\sum_{j=0}^{2n}\varphi^j\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}\right)\\ &= \sum_{i,j=0}^{2n}\varphi^{i+j}\int_{\hat{\Omega}}\nabla(\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{u}}) : \nabla(\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}})\\ &\quad+\sum_{i,j=0}^{2n}\varphi^{i+j+1}\int_{\hat{\Omega}}\nabla(\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{u}}) : (\boldsymbol{D}_1\nabla)(\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}})\\ &\quad-\sum_{i,j=0}^{2n}\varphi^{i+j+2}\int_{\hat{\Omega}}\nabla(\boldsymbol{B}_i^{(+)}\hat{\boldsymbol{u}}) : (\boldsymbol{D}_2\nabla)(\boldsymbol{B}_j^{(+)}\hat{\boldsymbol{w}}),\end{aligned} \tag{82}$$

which is an affine representation with $4n+2$ terms.

6. Results

For both the Taylor–Hood and the conforming method an ensemble of 225 solutions was generated at the 15×15 Gauss points for the parameter set

$$\mathcal{P} = \{(\varphi, u_\infty) \mid \varphi\in[-35^\circ, 35^\circ], u_\infty\in[1,20]\}.$$

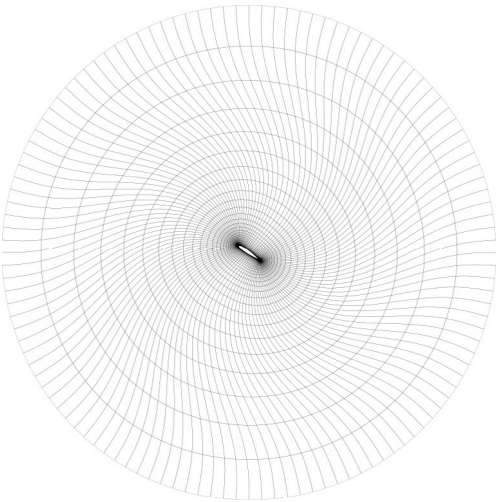

Fig. 3. Sample domain with $\varphi = -\pi/4$.

The viscosity was set at $\nu = 1/6$. The airfoil was chosen as a NACA0015 profile with a chord length of 1, thus giving an approximate maximal Reynold's number of 120.

The reference domain $\hat{\Omega}$ was chosen as a disk of radius 10, centered at the center of the airfoil, excluding the airfoil at $\varphi = 0°$. The mesh had 120 elements in the circumferential direction and 40 elements in the radial direction. Given discretization nodes $\{n_i^{\mathrm{INT}}\}_{i=1}^{120}$ for the airfoil, and uniformly spaced nodes $\{n_i^{\mathrm{EXT}}\}_{i=1}^{120}$ for the external boundary, the interior nodes of the mesh were positioned as

$$n_{ij}^{\mathrm{DOM}} = \left(\frac{e^{\gamma j/120} - 1}{e^{\gamma} - 1}\right) n_i^{\mathrm{INT}} + \left(1 - \frac{e^{\gamma j/120} - 1}{e^{\gamma} - 1}\right) n_i^{\mathrm{EXT}} \tag{83}$$

where $\gamma = 120 \log \alpha$ and α is a grading factor. This produces a mesh where the elements grow in size by a factor of approximately α for each radial layer, starting from the airfoil. We used $\alpha = 1.2$.

The boundary conditions were enforced strongly at the inflow boundary, with no-slip conditions on the airfoil and a do-nothing homogeneous Neumann boundary condition on the outflow. The lift function was chosen as the solution to the Stokes problem with $\varphi = 0$, which is solenoidal.

Remark 3. For divergence-conforming B-spline discretizations, strong enforcement of essential (e.g. no-slip) boundary conditions across corners of the domain leads to loss of inf–sup stability [34]. To resolve this instability, either the pressure space must be reduced by nullifying it at the respective corners [34] or a weak enforcement of boundary conditions must be applied [36]. Let us note that the considered airfoil test case is devoid of corner instabilities by virtue of the fact that the domain is a topological annulus, i.e. a domain without corners.

For parametrizing the geometry, the radius-dependent rotation angle function θ was chosen as

$$\theta(r) = \begin{cases} 1, & r < r_{\min}, \\ 0, & r > r_{\max}, \\ (1-\overline{r})^3(3\overline{r}+1), & \text{otherwise}, \end{cases} \tag{84}$$

with

$$\overline{r} = \frac{(r - r_{\min})}{(r_{\max} - r_{\min})}, \qquad r_{\min} = 1, \qquad r_{\max} = 10,$$

see Figs. 2 and 3.

For the conventional high-fidelity model, we chose a Lagrangian bi-quadratic basis for velocity, and Lagrangian bilinear basis for the pressure. This fulfills the Taylor–Hood property, leading to an inf–sup stable formulation. For the conforming high-fidelity model, we chose a bilinear Lagrangian basis for the pressure, and a mixed quadratic

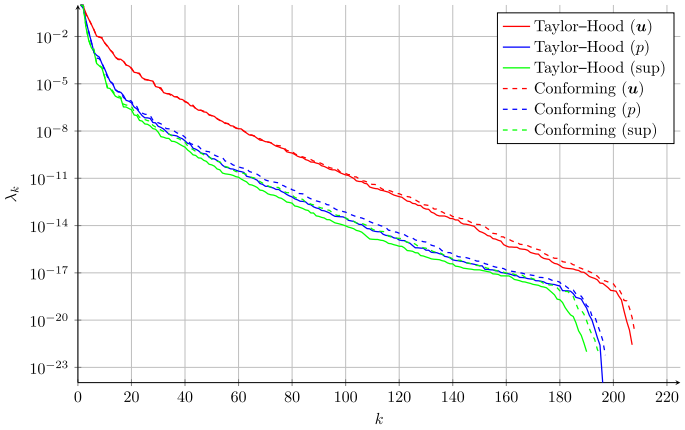

Fig. 4. Ensemble spectra for 225 ensemble solutions for both methods, and all three spaces (velocity, pressure and supremizers).

and linear B-spline basis for the velocity, giving a fully divergence-conforming method for the reference geometry at $\varphi = 0$ (see [36]). Note that by design, the divergence-conforming method will retain this property for other angles.

For solving the nonlinear equation we used Newton iteration, stopping when the velocity update reached 10^{-10} or less, as measured in the H^1 seminorm. The affine representations were derived from a truncated version of (71) with $n = 12$ terms, sufficient to represent the rotation matrix $\boldsymbol{R}(a)$ to 10 digits accuracy within the range of angles considered.

Results have been generated for reduced models with $M = 10, 20, \ldots, 50$ degrees of freedom each in the three spaces (velocity, supremizers and pressure), as well as the corresponding un-stabilized models (only velocity and pressure). Additionally, for the conforming stabilized method, a distinction is made between a naive solver, a block solver (see (65)–(66)) and a combined solver (using the pressure reconstruction technique in Section 4.2).

6.1. Spectrum

The decay rate of the eigenvalue spectrum of the solution ensembles indicates to which degree one might expect a reduced basis to adequately capture the most significant behavioral patterns of the model. As can be seen from Fig. 4, the decay is rapid for the first 20 or so modes, and then flattens out somewhat after that. (There is reason to believe that this might be improved for different choices of the angle function θ.) The spectra for the supremizers closely mimic those of the pressure solutions, as should be expected.

Remark 4. The difference in the pressure spectra can be partially explained by quadrature. We have used the same quadrature rule (9-point Gaussian quadrature) for both methods. However, the conforming method has higher order integrands than the Taylor–Hood method, and thus requires stronger quadrature rules. The relative quadrature error for the conforming method is estimated at roughly 10^{-4} for both velocity and pressure, while the same errors for the Taylor–Hood method are 10^{-5} and 10^{-6}, respectively.

6.2. Basis functions

The six dominant modes for velocity, pressure and supremizers are shown in Figs. 5–7 for the Taylor–Hood method and Figs. 8–10 for the conforming method. As is common with reduced basis methods, we can see that the first modes represent the most dominant flow patterns, and that it is left for the higher order modes to represent finer details, in this case wake effects. It is interesting that the first velocity modes of the two methods do not agree (in fact, the second mode of the conforming method agrees with the first mode of the Taylor–Hood method). The first two supremizer modes of the two methods also appear to be "switched". As for the pressure modes, aside from irrelevant sign flips, they are in good agreement between the two methods, which is natural considering that the function space mapping for the pressure is identical.

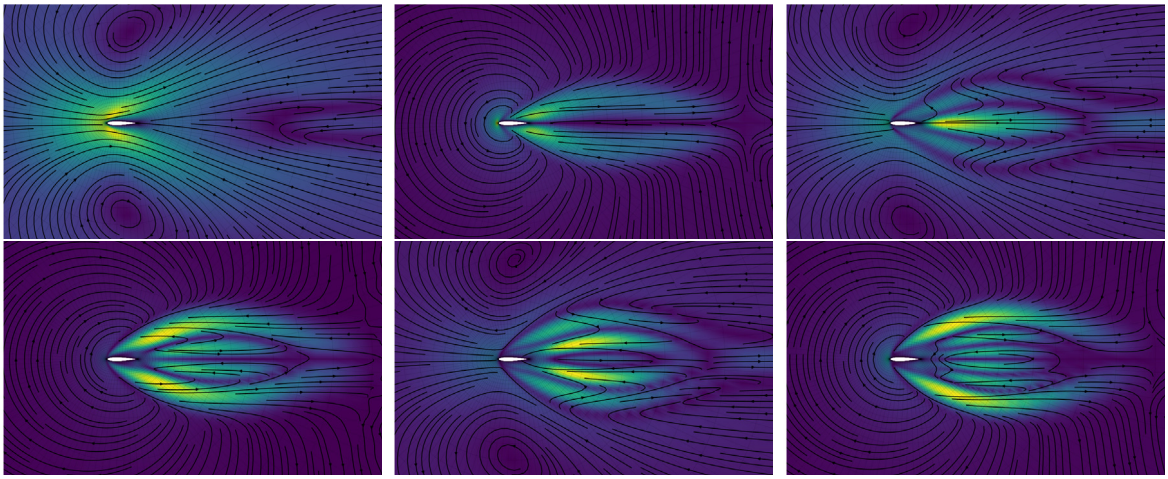

Fig. 5. First six velocity modes (Taylor–Hood method).

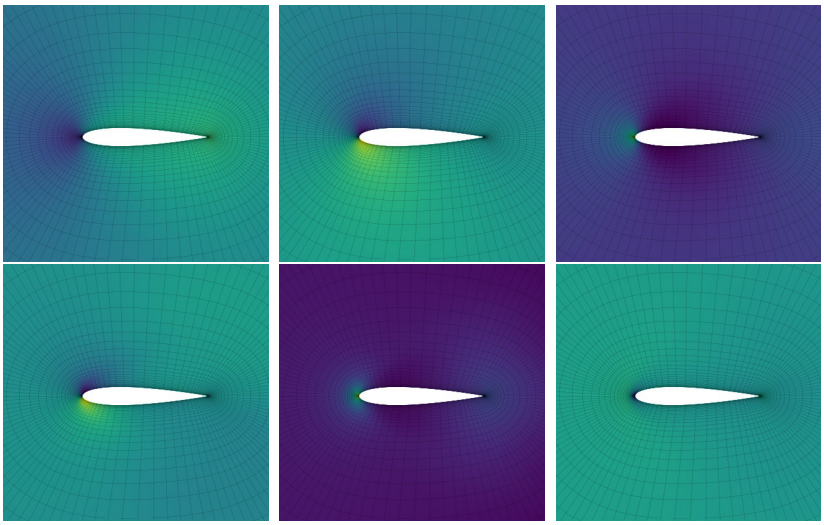

Fig. 6. First six pressure modes (Taylor–Hood method).

The divergence of the velocity basis functions is shown in Fig. 11. Here, the *mean* and *maximal* divergence of the ten first basis functions for each method (measured in the L^2-norm) are shown for various angles of attack. The figure shows that the divergences of the basis functions for the conventional method range between approximately 1 and 10. The divergences of the reduced-basis functions for the conforming method are consistently zero to 11 or 12 digits of accuracy (well within the 10 digits chosen as a baseline when choosing the number of terms n.) More importantly, it reveals that the solenoidality property is consistently satisfied throughout the parameter domain, with no significant variability.

6.3. Stability

To understand the stability properties of the four methods, viz. stabilized and unstabilized Taylor–Hood and conforming, we computed the LBB constants β_h from (53) according to (56). The results are presented in Table 1. The results clearly demonstrate the ability of the supremizer enrichment procedure to stabilize the method. In addition, the results highlight that no significant degradation of stability is to be expected for divergence-conforming reduced methods.

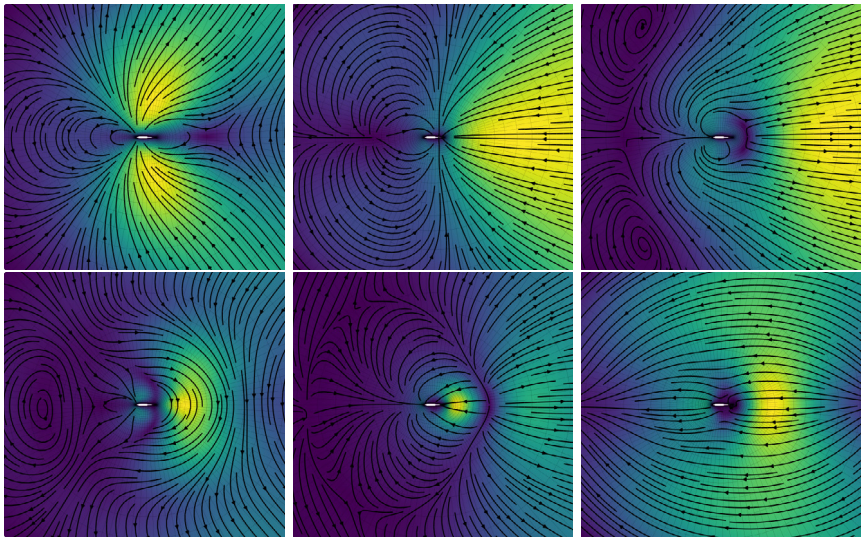

Fig. 7. First six supremizer modes (Taylor–Hood method).

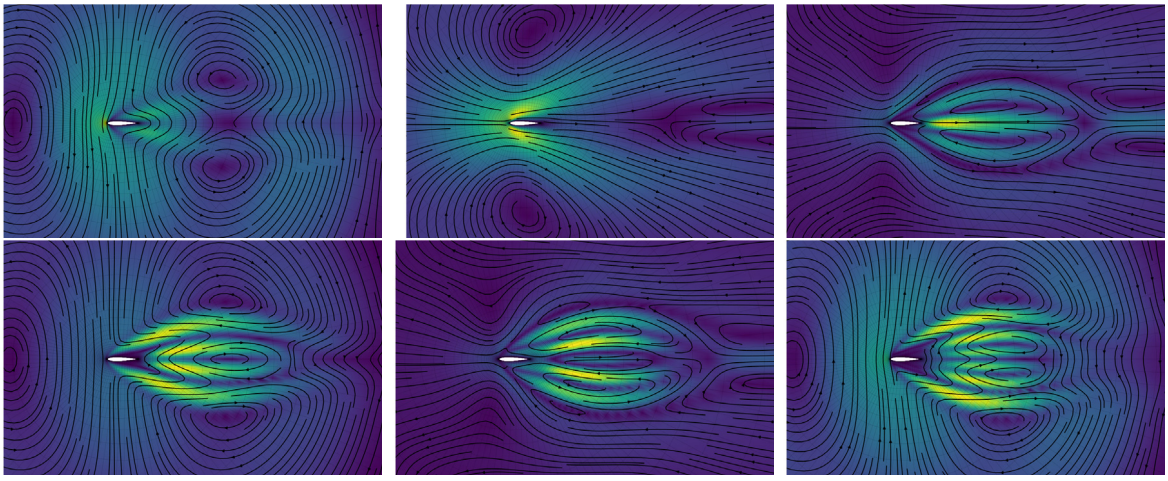

Fig. 8. First six velocity modes (conforming method).

Table 1
LBB constants β_h, as defined by (53), for various basis sizes and methods. The reported values are *minima* over a sampling of 15 × 15 parameter values. The LBB constants for the unstabilized conforming method were all indistinguishable from zero to machine precision.

	Taylor–Hood		Conforming	
♯ DoFs (M)	Unstabilized	Stabilized	Unstabilized	Stabilized
10	8.06×10^{-6}	3.93×10^{-1}	2.68×10^{-18}	4.11×10^{-1}
20	5.68×10^{-6}	2.54×10^{-1}	4.05×10^{-18}	3.37×10^{-1}
30	1.03×10^{-6}	3.15×10^{-1}	1.96×10^{-19}	3.13×10^{-1}
40	2.65×10^{-8}	2.67×10^{-1}	1.69×10^{-18}	2.90×10^{-1}
50	1.77×10^{-9}	2.75×10^{-1}	3.17×10^{-18}	2.72×10^{-1}

Table 2 shows the minimal principal angles between the velocity and supremizer spaces, computed according to [58]. Since linear independence between the two is not enforced *a priori*, these angles are relevant to the question

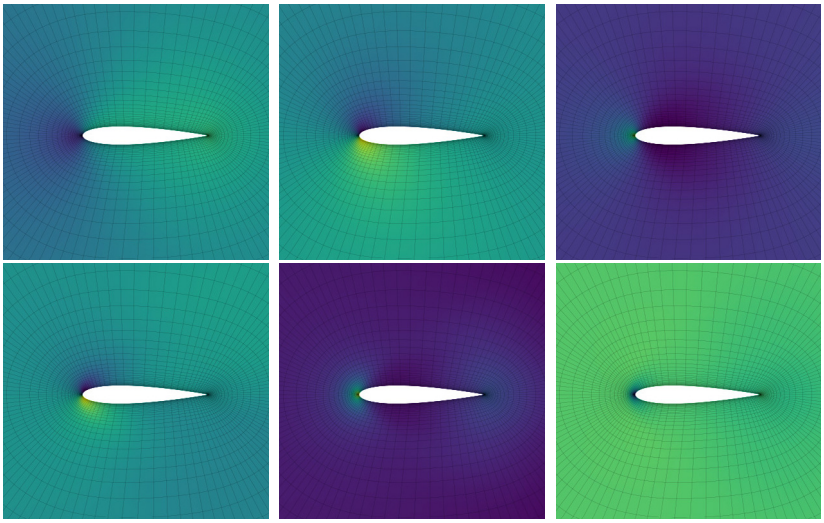

Fig. 9. First six pressure modes (conforming method)

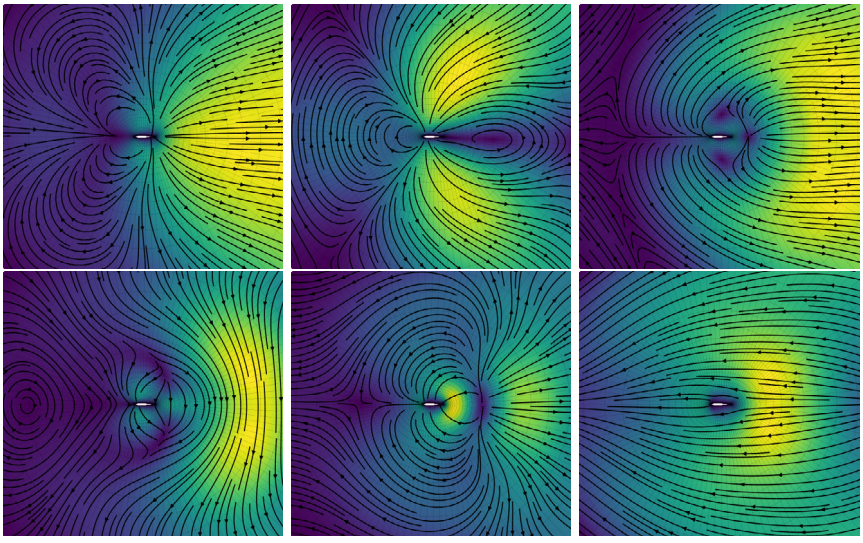

Fig. 10. First six supremizer modes (conforming method)

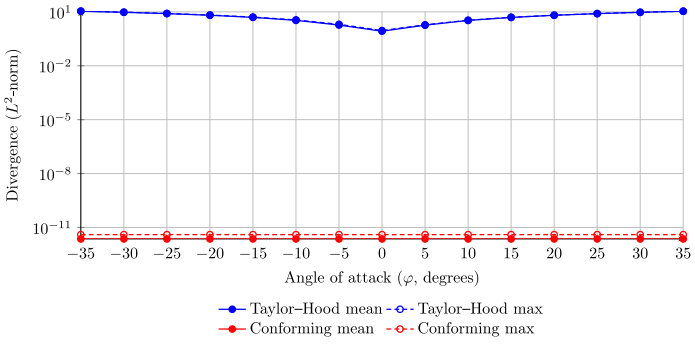

Fig. 11. Mean and maximal divergences of the ten first basis functions for the Taylor–Hood and conforming methods, as measured in L^2-norms, for various angles of attack.

Table 2
Minimal principal angles between the velocity and supremizer spaces for the two methods for various reduced basis sizes.

♯ DoFs (M)	Taylor–Hood	Conforming
10	82.2°	89.9°
20	75.4°	88.9°
30	65.7°	88.3°
40	62.7°	86.9°
50	59.4°	85.7°

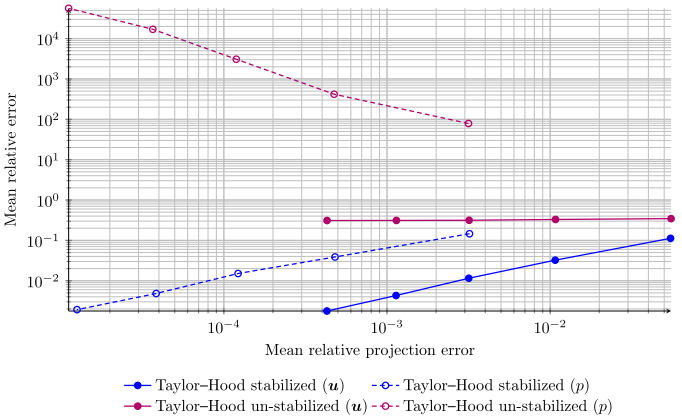

Fig. 12. Measured mean relative error as a function of mean relative projection error, for velocity (H^1-seminorm) and pressure (L^2-norm). The Taylor–Hood stabilized and un-stabilized methods are shown. Stabilization is clearly necessary to obtain any sort of useful solution.

whether the sum velocity space may possibly be degenerate. This shows that for all bases considered in this work, the two spaces are suitably linearly independent. It is noteworthy that the divergence-conforming bases show remarkable near-orthogonality for larger M that the Taylor–Hood method does not match.

6.4. Performance

Performance metrics are presented in terms of error, speedup and degrees of freedom.

- *Error*, when reported, is always given as *mean relative error* between the result of the high-fidelity and the reduced basis method. The mean is taken over $225 = 15 \times 15$ uniformly spaced points in the parameter space. The error norms used are H^1 seminorm for velocities and L^2 norm for pressure. For the unstabilized reduced methods, we found that they consistently failed to converge, thus the errors reported for those methods are based on fewer parameter values (those where convergence could be achieved).
- *Projection error* is, given M, the smallest ϵ satisfying (44). We used prescribed values of $M = 10, 20, 30, 40, 50$, computing ϵ as a function of M rather than the other way around.
- *Time usage* is the mean time spent solving one instance of the problem. This includes time spent in both the assembly and the linear solver phase, and notably will also be affected by the number of *iterations* the nonlinear solver needs to converge.
- *Degrees of freedom* is the number M of basis functions in a reduced space for a single field. For the Taylor–Hood method, this implies a total of $2M$ degrees of freedom in velocity (although only M of these can be expected to have approximative power, the supremizer functions are indispensable and *do* contribute to the final solution), and M degrees of freedom in pressure. For the conforming method, there are M degrees of freedom in both spaces.

Fig. 12 first shows measured error as a function of projection error, for the stabilized and unstabilized Taylor–Hood methods. While the stabilized method can be seen to converge, the velocity solution of the un-stabilized methods

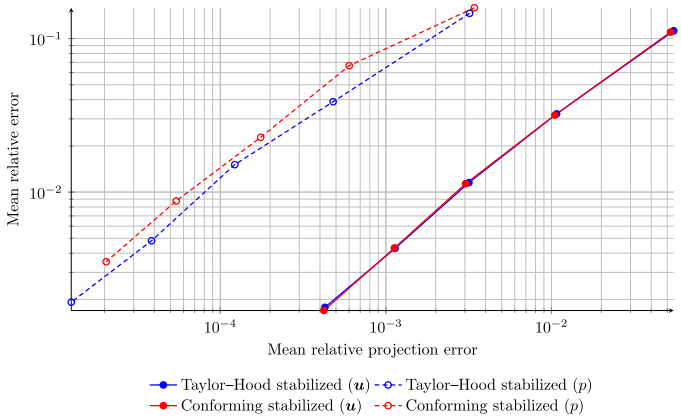

Fig. 13. Measured mean relative error as a function of mean relative projection error, for velocity (H^1-seminorm) and pressure (L^2-norm). Both Taylor–Hood and conforming are shown. The conforming block solver is indistinguishable from the naive conforming solver, and the velocity errors of the Taylor–Hood and the conforming methods are also virtually indistinguishable from each other. Bottom right is better.

does not, and the pressure solution diverges. In addition to this, the Taylor–Hood un-stabilized method has serious convergence problems in the nonlinear solver, especially for larger M, and the conforming un-stabilized methods essentially never converges for any parameter values. In the following we will therefore ignore the unstabilized methods.

Fig. 13 shows the same results, comparing the Taylor–Hood and the conforming methods (both stabilized). The conforming method obtains somewhat better velocity results and comparable pressure results to the Taylor–Hood method. For both the Taylor–Hood and the conforming approximations, the mean relative error is essentially proportional to the mean relative projection error, both for pressure and for velocity. For the velocity approximations, the actual mean relative error is approximately 3 times larger than the projection error. For the pressure approximations, the mean relative error is approximately 100 times larger than the projection error. To elucidate the deviations between the actual errors and the projection errors, we note that for Galerkin subspace approximations $\boldsymbol{u}^{\mathrm{R}} \in U^{\mathrm{R}} \subset [H^1(\Omega)]^d$, $p^{\mathrm{R}} \in P^{\mathrm{R}} \subset L^2(\Omega)$ of a well-posed mixed linear problem of the generic form:

$$
\begin{aligned}
&Find\ (\boldsymbol{u}, p) \in [H^1(\Omega)]^d \times L^2(\Omega):\\
&\qquad a(\boldsymbol{u}, \boldsymbol{w}) + b(p, \boldsymbol{w}) = d(\boldsymbol{w}) \qquad\qquad \forall \boldsymbol{w} \in [H^1(\Omega)]^d\\
&\qquad b(q, \boldsymbol{u}) = 0 \qquad\qquad \forall q \in L^2(\Omega)
\end{aligned}
$$

the following a-priori estimates hold:

$$
\|\boldsymbol{u} - \boldsymbol{u}^{\mathrm{R}}\|_{H^1(\Omega)} \leq \left(1 + \frac{c_a}{\beta_a}\right) \inf_{v^{\mathrm{R}} \in U^{\mathrm{R}}} \|\boldsymbol{u} - \boldsymbol{v}^{\mathrm{R}}\|_{H^1(\Omega)} \tag{85}
$$

$$
\|p - p^{\mathrm{R}}\|_{L^2(\Omega)} \leq \left(1 + \frac{1}{\beta_b^{\mathrm{R}}}\right) \inf_{q^{\mathrm{R}} \in P^{\mathrm{R}}} \|p - q^{\mathrm{R}}\|_{L^2(\Omega)} + \frac{c_a}{\beta_b^{\mathrm{R}}} \|\boldsymbol{u} - \boldsymbol{u}^{\mathrm{R}}\|_{H^1(\Omega)} \tag{86}
$$

Herein c_a and β_a denote the continuity and coercivity constants of the bilinear forms a, respectively, and β_b^{R} is the inf–sup constant of the bilinear form b in the subspace $U^{\mathrm{R}} \times P^{\mathrm{R}}$. Estimate (85) follows from Céa's lemma. For (86) see for instance [37,59]. The projection errors in Fig. 13 correspond to the minimal errors that can be achieved in the RB spaces. Hence, the projection errors coincide with the infima that appear in the right members of (85)–(86). The left members of (85)–(86) pertain to the actual RB Galerkin-approximation errors. From (85)–(86) one can infer that equivalence between the mean relative error and the mean relative projection error in Fig. 13 can only be asserted up to a proportionality constant that depends on the stability of the problem. Moreover, estimate (86) conveys that the RB Galerkin approximation of the pressure incurs an induced contribution from the velocity error. Noting that the error in the velocity is significantly higher than the projection error in the pressure (the difference being a factor of approximately 90), the induced error in the pressure can cause a notable deviation between the projection error and the actual error. For the LBB constants in Table 1, the factor $(1 + 1/\beta_b^{\mathrm{R}})$ in the first term on the right-hand side of (86)

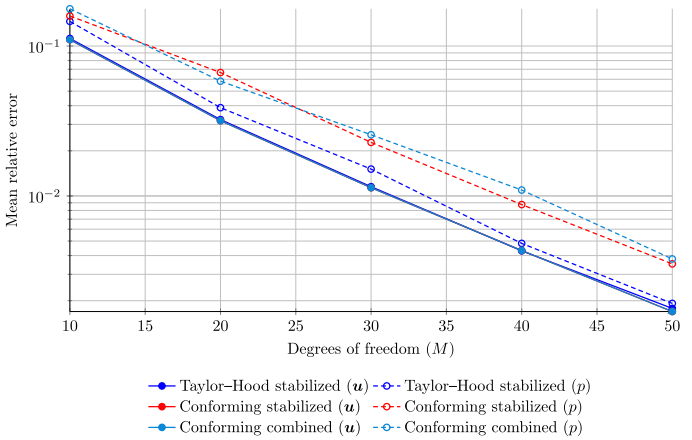

Fig. 14. Measured mean relative error as a function of M, for velocity (H^1-seminorm) and pressure (L^2-norm). Both Taylor–Hood and conforming are shown. The velocity errors for the two conforming methods overlap. Bottom left is better.

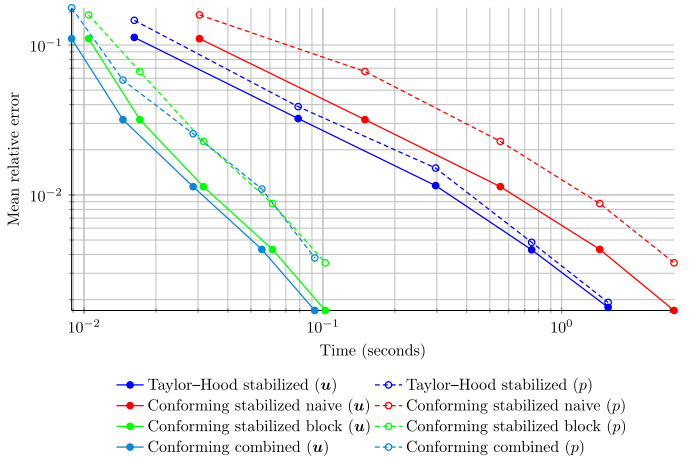

Fig. 15. Measured mean relative error as a function of mean time usage (in seconds), for velocity (H^1-seminorm) and pressure (L^2-norm). The time reported includes both the velocity solution and the pressure recovery or reconstruction, as appropriate. Both Taylor–Hood and conforming are shown, the latter with two different solvers (the naive solver and the block solver), as well as with the joint velocity–pressure reconstruction scheme. Bottom left is better.

is ≤ 5. The remaining predominant part of the difference between the mean relative projection error and the mean relative error in the reduced pressure approximation is accounted for by the second term in right member.

Fig. 14 shows error as a function of degrees of freedom. Here, it is clear that the pressure results for the conventional method are slightly better *per degree of freedom*. The difference in the pressure errors is more pronounced than in Fig. 13 because part of the discrepancy can be attributed to corresponding differences in the spectrum; cf. Fig. 4. That is to say, due to the slightly slower decay of eigenvalues in the conforming pressure snapshots as compared to the Taylor–Hood modes, the projection error for reduced models with the same number of degrees of freedom will be higher for the latter method. In Fig. 14 we can also see that the combined velocity–pressure basis (Section 4.2) has reconstructed pressure errors that are only slightly worse than the recovered pressures from Section 4.1. It is to be noted, however, that it is not evident that these good approximation properties of the reconstructed pressure will be retained for more strongly nonlinear cases.

Fig. 15 shows the measured errors as a function of mean time usage, measured in seconds. The time usage reported is the total time taken for both velocity solution and pressure recovery, as appropriate. This clearly shows the ability of the conforming method to achieve the same results significantly faster if the block solver algorithm from (65)–(66) is employed, taking advantage of the block triangular form of the linear system. It also shows how the combined

velocity–pressure basis is able to achieve the same velocity results slightly faster, on account of not requiring an additional linear system to solve for the pressure.

All performance data were collected on an Intel Core Skylake i7-6700K system operating at 4 GHz, using linear solvers from Numpy 1.14.2 (dense systems) and Scipy 1.0.1 (sparse systems) compiled by Anaconda with bindings to Intel's MKL library.

7. Conclusions

We have formulated the fundamentals for Reduced Basis Methods (RBMs) for stationary Navier–Stokes problems. Using a divergence-conforming isogeometric high-fidelity method, we were able to generate divergence-free reduced bases, yielding a velocity-only reduced formulation. To solve for pressure, we employed supremizers as test functions, a technique used to stabilize conventional reduced basis methods. This allowed us to achieve accurate reduced solutions for both velocity and pressure with significant speed benefits over a coupled conventional formulation. In addition, we considered the option of direct pressure reconstruction, using the coefficients of the RB velocity solution.

Investigations were performed for the parametrized problem of flow around a NACA0015 airfoil with varying angle-of-attack and inflow velocity. The divergence-conforming method was compared with a supremizer stabilized reduced method based on the Taylor–Hood high-fidelity scheme with the same mesh and comparable polynomial degree. For the divergence-conforming method, a velocity-only formulation was applied, with two choices for the pressure approximation, viz. recovery and reconstruction.

While the accuracy of the reduced methods was virtually identical, the divergence-conforming models were able to achieve a significantly faster online stage thanks to the velocity–pressure decoupling.

The pressure recovery and reconstruction methods were also able to achieve similar performance for the considered problem. Since the pressure reconstruction method relies on a linearity assumption that is not valid in general, we must conclude that pressure recovery is to be preferred in most cases.

The faster online stage of the conforming RB method is offset by a more complicated offline stage. In particular, the affine representations (29)–(34) of the divergence-conforming method required approximately three times as many terms ($\sim 12n$ versus $\sim 4n$). In spite of this, the conforming method yielded significantly faster online stages. We aim to investigate in future work in which manner automatic "black box" methods such as Empirical Interpolation (EIM, see [55–57] and [19, Chapter 10]) can alleviate the additional work needed to produce such affine representations.

We will also investigate the performance of the present approach when using adaptive high-fidelity models based on *a posteriori* error estimates developed in [60,61].

In the present work we restricted our considerations to stationary flows at low Reynold's numbers. The main idea of using solenoidal snapshots in a velocity-only formulation is generic and carries over to more complicated flow problems, e.g. three dimensional problems, time-dependent flows and high Reynold's number flows. For high Reynold's number applications, the variational formulation needs to be enhanced with additional stabilization terms, e.g. with the Streamwise Upwind Petrov Galerkin (SUPG) formulation. Such a modification will carry through to the RB method, specifically in the affine representations. In addition, the treatment of nonlinear terms by explicit computation and storage of representative multi-linear forms will not be feasible for large scale problems. The presented RB methodology can be extended to time-dependent problems in the usual manner, i.e. by collecting snapshots at different time steps in the offline stage and use these to construct a reduced basis for the time integration in the online stage.

Acknowledgments

The authors acknowledge the financial support from the Norwegian Research Council and the industrial partners of OPWIND: Operational Control for Wind Power Plants (Grant No.: 268044/E20).

References

[1] M.A. Bazaz, M. un Nabi, S. Janardhanan, A review of parametric model order reduction techniques, in: 2012 IEEE International Conference on Signal Processing, Computing and Control, 2012, pp. 1–6.

[2] F. Chinesta, P. Ladevèze, Separated representations and PGD-based model reduction, in: Fundamentals and Applications, International Centre for Mechanical Siences, Courses and Lectures, vol. 554, Springer, 2014.

[3] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer, Springer Science & Business Media, 2013.

[4] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395.
[5] A. Dumon, C. Allery, A. Ammar, Proper general decomposition (PGD) for the resolution of Navier–Stokes equations, J. Comput. Phys. 230 (4) (2011) 1387–1407.
[6] B. Almroth, P. Stern, F. Brogan, Automatic choice of global shape functions in structural analysis, AIAA J. 16 (5) (1978) 525–528.
[7] B. Almroth, P. Stehlin, F. Brogan, Use of global functions for improvement in efficiency of nonlinear analysis, in: 22nd Structures, Structural Dynamics and Materials Conference, 1981, p. 575.
[8] D.A. Nagy, Modal representation of geometrically nonlinear behavior by the finite element method, Comput. Struct. 10 (4) (1979) 683–688.
[9] A.K. Noor, J.M. Peters, Reduced basis technique for nonlinear analysis of structures, AIAA J. 18 (4) (1980) 455–462.
[10] A.K. Noor, J.M. Peters, Bifurcation and post-buckling analysis of laminated composite plates via reduced basis technique, Comput. Methods Appl. Mech. Engrg. 29 (3) (1981) 271–295.
[11] A.K. Noor, Recent advances in reduction methods for nonlinear problems, in: Computational Methods in Nonlinear Structural and Solid Mechanics, Elsevier, 1981, pp. 31–44.
[12] A.K. Noor, On making large nonlinear problems small, Comput. Methods Appl. Mech. Engrg. 34 (1–3) (1982) 955–985.
[13] J.P. Fink, W.C. Rheinboldt, On the error behavior of the reduced basis technique for nonlinear finite element approximations, ZAMM Z. Angew. Math. Mech. 63 (1) (1983) 21–28.
[14] J.P. Fink, W.C. Rheinboldt, Solution manifolds and submanifolds of parametrized equations and their discretization errors, Numer. Math. 45 (3) (1984) 323–343.
[15] M.D. Gunzburger, Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice, and Algorithms, Elsevier, 2012.
[16] J.S. Peterson, The reduced basis method for incompressible viscous flow calculations, SIAM J. Sci. Stat. Comput. 10 (4) (1989) 777–786.
[17] C. Prud'Homme, D.V. Rovas, K. Veroy, A.T. Patera, A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations, ESAIM Math. Model. Numer. Anal. 36 (5) (2002) 747–771.
[18] K. Veroy, C. Prud'Homme, D. Rovas, A. Patera, A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, in: 16th AIAA Computational Fluid Dynamics Conference, 2003, p. 3847.
[19] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, Springer International Publishing, 2016.
[20] B. Haasdonk, Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems, in: Model Reduction and Approximation: Theory and Algorithms, vol. 15, SIAM, 2017, p. 65.
[21] S.S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, Internat. J. Numer. Methods Fluids 34 (5) (2000) 425–448.
[22] K. Ito, S.S. Ravindran, Reduced basis method for optimal control of unsteady viscous flows, Int. J. Comput. Fluid Dyn. 15 (2) (2001) 97–113.
[23] G.R. Liu, J. Lee, A.T. Patera, Z.L. Yang, K.Y. Lam, Inverse identification of thermal parameters using reduced-basis method, Comput. Methods Appl. Mech. Engrg. 194 (27–29) (2005) 3090–3107.
[24] C. Lieberman, K. Willcox, O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, SIAM J. Sci. Comput. 32 (5) (2010) 2523–2542.
[25] A. Manzoni, A. Quarteroni, G. Rozza, Shape optimization for viscous flows by reduced basis methods and free-form deformation, Internat. J. Numer. Methods Fluids 70 (5) (2012) 646–670.
[26] G. Rozza, T. Lassila, A. Manzoni, Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map, in: Spectral and High Order Methods for Partial Differential Equations, Springer, 2011, pp. 307–315.
[27] G.S.H. Pau, Reduced Basis Method for Quantum Models of Crystalline Solids (Ph.D. thesis), Massachusetts Institute of Technology, 2007.
[28] P. Krysl, S. Lall, J.E. Marsden, Dimensional model reduction in non-linear finite element dynamics of solids and structures, Internat. J. Numer. Methods Engrg. 51 (4) (2001) 479–504.
[29] A. Iollo, S. Lanteri, J.A. Désidéri, Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations, Theor. Comput. Fluid Dyn. 13 (6) (2000) 377–396.
[30] G. Stabile, G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations, 2017, arXiv preprint arXiv:1710.11580.
[31] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1136–1161.
[32] A.E. Løvgren, Y. Maday, E.M. Rønquist, A reduced basis element method for the steady Stokes problem, ESAIM Math. Model. Numer. Anal. 40 (3) (2006) 529–552.
[33] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Comput. Methods Appl. Mech. Engrg. 199 (17–20) (2010) 1143–1152.
[34] A. Buffa, C. De Falco, G. Sangalli, Isogeometric analysis: Stable elements for the 2D Stokes equation, Internat. J. Numer. Methods Fluids 65 (11–12) (2011) 1407–1422.
[35] J.A. Evans, T.J.R. Hughes, Discrete spectrum analyses for various mixed discretizations of the Stokes eigenproblem, Comput. Mech. 50 (6) (2012) 667–674.
[36] J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier-Stokes equations, Math. Models Methods Appl. Sci. 23 (8) (2013) 1421–1478.
[37] J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the Darcy-Stokes-Brinkman equations, Math. Models Methods Appl. Sci. 23 (04) (2013) 671–741.
[38] J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations, J. Comput. Phys. 241 (2013) 141–167.
[39] T. Dokken, T. Lyche, K.F. Pettersen, Polynomial splines over locally refined box-partitions., Comput. Aided Geom. Design 30 (3) (2013) 331–356.

[40] K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines, Comput. Methods Appl. Mech. Engrg. 269 (2014) 471–514.
[41] K.A. Johannessen, M. Kumar, T. Kvamsdal, Divergence-conforming discretization for Stokes problem on locally refined meshes using LR B-splines, Comput. Methods Appl. Mech. Engrg. 293 (2015) 38–70.
[42] T.M. van Opstal, J. Yan, C. Coley, J.A. Evans, T. Kvamsdal, Y. Bazilevs, Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows, Comput. Methods Appl. Mech. Engrg. 316 (2017) 859–879.
[43] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, Wiley, 2009.
[44] H. Melbø, T. Kvamsdal, Goal oriented error estimators for Stokes equations based on variationally consistent postprocessing, Comput. Methods Appl. Mech. Engrg. 192 (5–6) (2003) 613–633.
[45] E.H. van Brummelen, K.G. van der Zee, V.V. Garg, S. Prudhomme, Flux evaluation in primal and dual boundary-coupled problems, J. Appl. Mech. 79 (1) (2012) 010904.
[46] C. Taylor, P. Hood, A numerical solution of the Navier–Stokes equations using the finite element technique, Internat. J. Numer. Methods Fluids 1 (1973) 73–100.
[47] J.r. Nédélec, A new family of mixed finite elements in $\mathbb{R}^3$, Numer. Math. 50 (1986) 57–81.
[48] P.A. Raviart, J.M. Thomas, Mathematical aspects of the finite element method, in: Lecture Notes in Mathematics, vol. 606, Springer, 1981, pp. 292–315.
[49] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Sci. (2000) 808–817.
[50] J. Weller, E. Lombardi, M. Bergmann, A. Iollo, Numerical Methods for Low-Order Modeling of Fluid Flows Based On POD, Research Report RR-6758, INRIA, 2008.
[51] M. Bergmann, C.H. Bruneau, A. Iollo, Enablers for robust POD models, J. Comput. Phys. 228 (2) (2009) 516–538.
[52] I. Akhtar, A.H. Nayfeh, C.J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows, Theor. Comput. Fluid Dyn. 23 (3) (2009) 213–237.
[53] A. Caiazzo, T. Iliescu, V. John, S. Schyschlowa, A numerical investigation of velocity-pressure reduced order models for incompressible flows, J. Comput. Phys. 259 (2014) 598–616.
[54] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, Internat. J. Numer. Methods Fluids 72 (12) (2013) 1219–1243.
[55] M. Barrault, Y. Maday, N. Nguyen, A. T. Patera, An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematique 339 (2004) 667–672, 11.
[56] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM Math. Model. Numer. Anal. 41 (3) (2007) 575–605.
[57] Y. Maday, N.C. Nguyen, A.T. Patera, G.S.H. Pau, A general, multipurpose interpolation procedure: the magic points, working paper or preprint, September 2007.
[58] A.V. Knyazev, M.E. Argentati, Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates, SIAM J. Sci. Comput. 23 (6) (2002) 2008–2040.
[59] T. Hoang, C.V. Verhoosel, F. Auricchio, E.H. van Brummelen, A. Reali, Mixed isogeometric finite cell methods for the Stokes problem, Comput. Methods Appl. Mech. Engrg. 316 (2017) 400–423.
[60] M. Kumar, T. Kvamsdal, K.A. Johannessen, Simple a posteriori error estimators in adaptive isogeometric analysis, Comput. Math. Appl. 70 (7) (2015) 1555–1582.
[61] M. Kumar, T. Kvamsdal, K.A. Johannessen, Superconvergent patch recovery and a posteriori error estimation technique in adaptive isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 316 (2017) 1086–1156.